Aspects of fractal image compression

Wakefield, Paul D.

Award date:
1999

Awarding institution:
University of Bath

ASPECTS OF FRACTAL IMAGE COMPRESSION

by

Paul D. Wakefield

A thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

University of Bath

1999

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

University of Bath

Abstract

ASPECTS OF FRACTAL IMAGE COMPRESSION

by Paul D. Wakefield

This work extends research on fractal image compression, which began with the proposal of the subject by Barnsley and the invention of block-based fractal coding by Jacquin. Later work by Monro and others created the Bath Fractal Transform (BFT) which improved compression performance by using a more complex basis.

Using the most recently published results this thesis continues development of the BFT by studying the effects of using an orthogonalisation operator at the parent mapping stage, improving the quantization of the transform parameters, and comparing methods of rate-distortion switching. In each of these areas useful observations and discoveries are made.

Later, a novel technique is introduced which can determine a parent location from the basis coefficients in a child block. This is named the Implicit Fractal Technique (IFT) and leads to an improvement in rate-distortion performance. As well as a complete derivation of the method and a comparison of its effects on different images, an attempt is made to extend the method by using a dual edge model.

Further work compares the results of zooming with several different fractal transforms, and improves zoomed image fidelity when using the IFT. A new fractal transform is also proposed which uses a generalised form of self-similarity, and has the potential to give greater fidelity and offers greater flexibility for transform design than existing methods. Finally methods of post-processing IFT coded images are investigated.

TABLE OF CONTENTS

Acknowledgements

This work was only made possible by the help and support of others during the period of my study at the University of Bath. In particular I would like to thank Donald Monro for supervising this project and securing the necessary financial support, and I would like to acknowledge how his enthusiasm and ideas were a vital source of encouragement and inspiration during my studies.

Financial support was provided by the University of Bath in the form of a studentship, and also by News Digital Systems (NDS). I am indebted to Arthur Mason of NDS for his help in arranging this backing and for his continued interest in my work. Additionally I am grateful for the provision of a stimulating work placement at the NDS offices in Southampton during the second year of my studies.

Within the Video Coding Group, I would like to thank fellow students David Bethel and Gary Dickson who have both provided ideas, advice, and help on numerous occasions. Other members of the Video Coding Group made the long and difficult process of writing up a great deal easier through their companionship. Finally I would like to thank Rosemary Ainsworth whose help in organising trips and other aspects of my course of study has been invaluable.

Statement of Originality

The author considers the following elements of this work to form an original contribution to the fractal image compression literature:

Chapter 2

- Literature review

Chapter 3

- Alternative mathematical description of the BFT

Chapter 5

- Discovery that orthogonalisation leads to rate-distortion improvement

Chapter 6

- Determination of optimal quantization parameters for BFT
- Introduction of novel automatic quantization method

Chapter 7

- Comparison of rate-distortion switching methods

Chapter 8

- Invention and mathematical derivation of Implicit Fractal Technique
- Evaluation of IFT

Chapter 9

- Development and evaluation of dual edge model

Chapter 10

- Comparison of partitioning criteria for orthogonal IFT

Chapter 11

- Proposal for new fractal transform

Chapter 12

- Comparison of fractal transforms for fractal zooming and resolution enhancement
- Improvement of IFT zooming results

Chapter 13

- Comparison of post-filtering methods

Appendix A

- Publications relating to this work

Appendix B

- Derivation of implicit parent location

Appendix C

- Derivation of extended implicit parent method

Overview

The main contribution of this work is the development of fractal methods for image compression. Initially the Bath Fractal Transform (BFT), a fractal image compression method invented and developed by Monro and co-workers, is adopted and the current state of the art reviewed.

A series of investigations into improvements to the BFT are presented, including the use of an orthogonalisation operator during the parent mapping step, truly optimal quantization of transform coefficients, and a rate-distortion switching technique where fractal terms are used selectively. These methods result in an algorithm which gives substantially improved fidelity.

Subsequently a new Implicit Fractal Technique (IFT) is derived which uses a mathematical model to estimate fractal properties of the image. Fidelity is improved in edge blocks on an image by exploiting a correlation between the basis coefficients of the transform and the location of the parent block relative to each child.

Later chapters present a mixture of work associated with BFT and IFT development. These include a comparison of possible quadtree partitioning criteria, a proposal for a new fractal transform which increases the fidelity which can be potentially achieved by fractal coding, a study of the multi-resolution property of fractal compression, and post-processing of fractal coded images.

The aim of this work throughout is to improve the performance of fractal transforms relative to other image compression methods. In particular, care is taken to focus on increasing the role of the fractal aspects of methods rather than the non-fractal components.

Parts of the work contained herein have been published elsewhere. Appendix A gives details and contains reproductions of several conference papers.

Chapter 1

INTRODUCTION

This chapter introduces four topics of relevance to this thesis. The first of these is fractals: complex sets which possess detail at all scales and are often strikingly beautiful, together with the mathematical tools of fractal geometry for analysing them. Secondly, a branch of fractal research entitled Iterated Function System theory, which facilitates the generation of fractal sets from very simple formulae. Thirdly, image compression, which is the conversion of image data into a more compact and efficient form. Finally, the principles of how *fractal* image compression may be possible.

1.1 Fractals

Due to the relative novelty of the subject no strict definition of "fractal" is widely used. An 'expansionist' definition has been in use since Mandelbrot's original essays [43,44] introduced fractals to the wider world. The reader is referred to these for a more complete, heuristic definition including a wide range of examples from both mathematics and the physical sciences. In a sense the specification is analogous to the characteristics of life [18] - suitably vague.

In the physical world a fractal object is something which has detail at such a range of scales as to render a classical geometric description impossible. In the physical world there are any number of such shapes: mountains, clouds, the path of lightning etc. As each is inspected more closely, more complex detail is revealed.

Take for example maps of the British coastline [44]. At a scale of 1/100,000 many bays and peninsulas are visible. When the map is compared with one of

scale 1/10,000 sub-bays and sub-peninsulas become visible. On a 1/1,000 map sub-sub-bays and sub-sub-peninsulas become visible.

In mathematics a fractal set is one with detail at every scale. One classic example is that of the Mandelbrot set, Figure 1.1, defined as the set of points c in the complex plane for which the Julia set of $f(z) = z^2 + c$ is connected.

The Julia set of a complex polynomial is defined as the closure of its set of repelling periodic points, so that for all points in the exterior, $|f^k(z)| \to \infty$.

From the formula itself one would not guess that both the Julia sets and the Mandelbrot set would have boundaries too complex to analyse with traditional geometry.

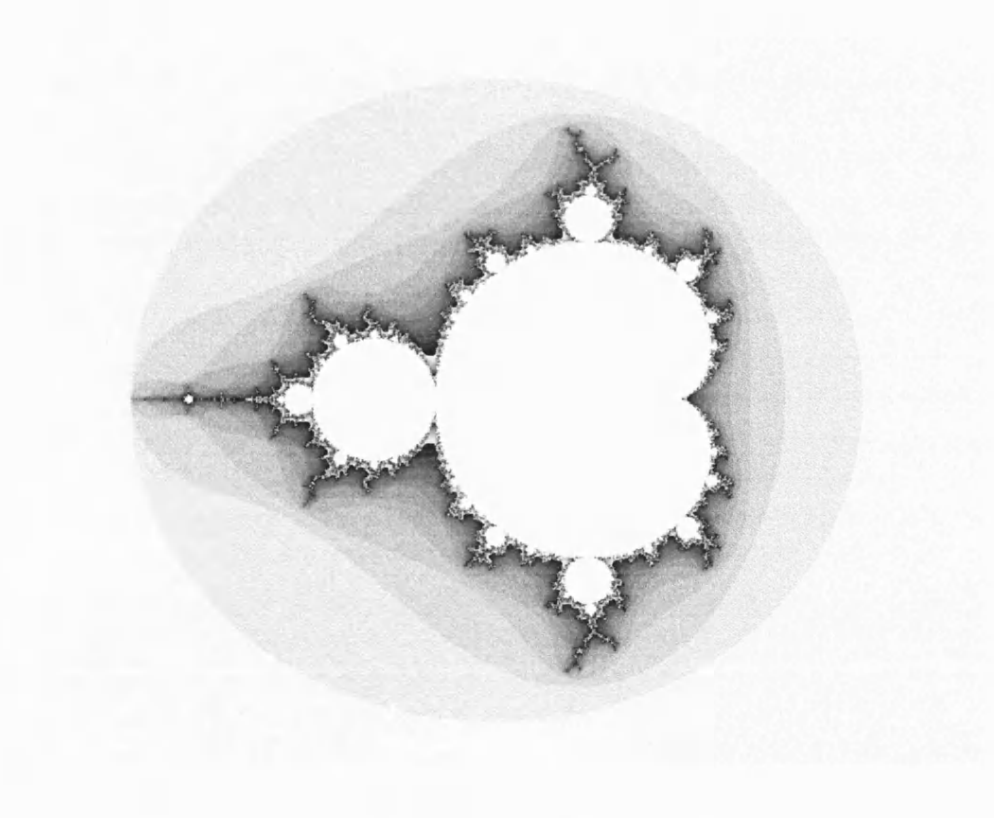


Figure 1.1. Mandelbrot Set. White points are members of the set.

Yet take one area of the boundary of the set and re-render it at a higher resolution. More detail can be observed. Repeat this action any number of times and at each stage more detail appears. Some example results are shown in Figure 1.2.

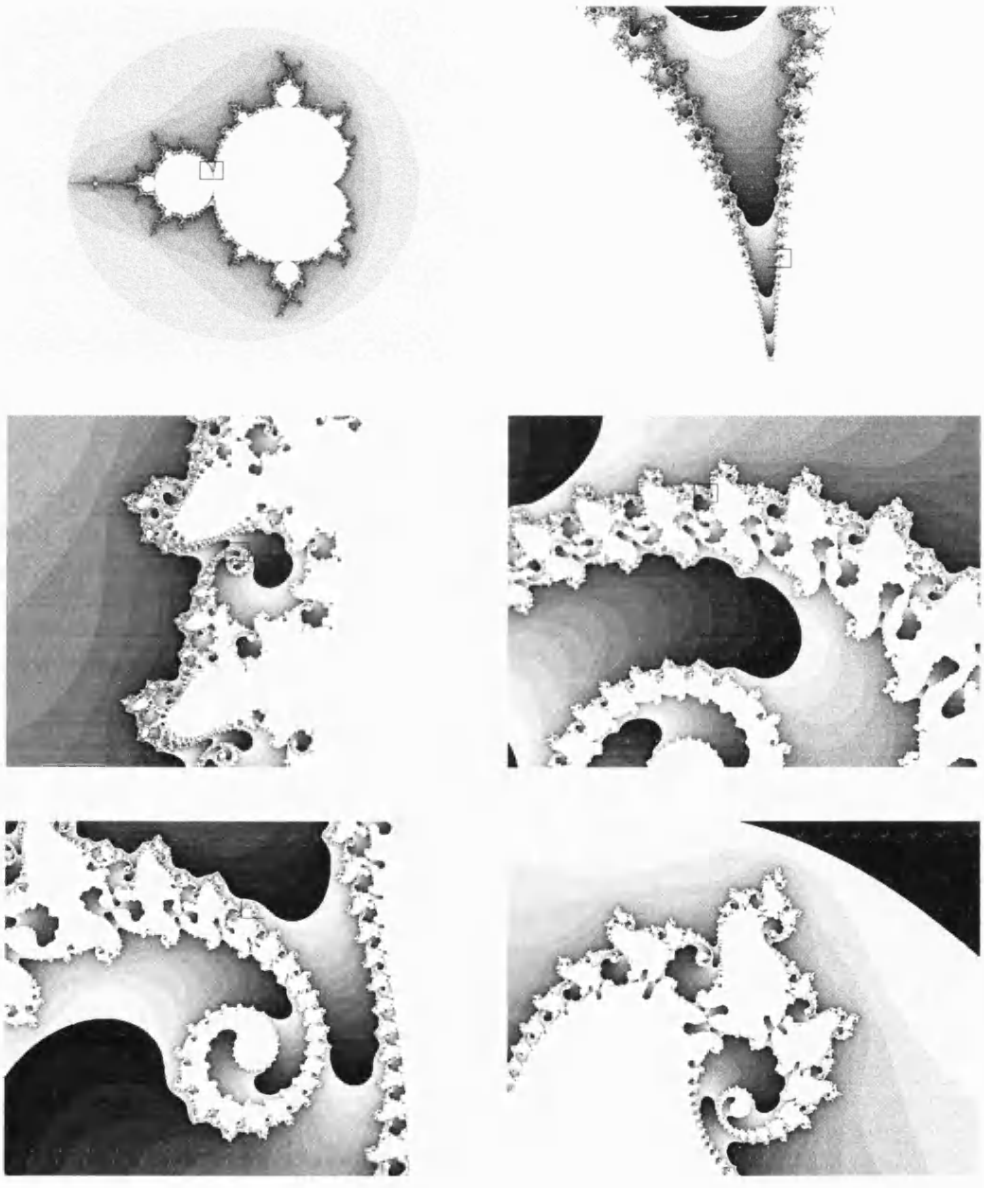


Figure 1.2 'Zoom' of the Mandelbrot Set at x20, x400, x8000, x160000, x3200000 magnification.

To facilitate the study of fractals it has been necessary to create a "fractal geometry". First among the tools used to analyse fractals is fractal dimension [18] - in simple terms an index of complexity. Using fractal dimension describes sets as n-dimensional where n is typically non-integer for fractal sets and integer for 'Euclidean' sets. Several different definitions exist, including the Hausdorff-Besicovitch dimension: the critical value $d = \dim_H F$ where $H^s(F) = \infty$ if $s < d$ and $H^s(F) = 0$ if $s > d$. Where $H^s(F)$ is the s-dimensional Hausdorff measure of F:

$$H^s(F) = \liminf_{\delta \to 0} \left\{ \sum_{i=1}^{\infty} |U_i|^s : \{U_i\} \text{ is a } \delta\text{-cover of } F \right\} \tag{1.1.1}$$

and the box-counting dimension:

$$d = \dim_B F = \lim_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta}, \tag{1.1.2}$$

where $N_\delta(F)$, is the smallest number of sets of diameter at most δ which can cover F, which can be applied to empirical data.

1.2 Iterated Function Systems

An Iterated Function System (IFS) [4,5,34] is a collection of contraction mappings on a complete metric space K, whose actions on subsets of K are combined to produce a contractive set-valued mapping. The result: An IFS possesses a unique attractor set which is typically fractal.

1.2.1 Theory

Let (K,d) be a compact metric space. For example, K may be the disk $\{x \in \mathbf{R}^2 : |x| \le 1\}$ and the distance function may be $d(x,y) = |x - y|$ in $\mathbf{R}^2$. Let $H(K)$ denote the set of all nonempty closed subsets of K, then it may be shown that $H(K)$ is a compact metric space with the Hausdorff metric

$$h(A,B) = \sup_{x\in A}\inf_{y\in B} d(x,y) + \sup_{x\in B}\inf_{y\in A} d(x,y) \tag{1.2.1}$$

for A, B subsets of K. (NB, a slightly different definition in terms of δ-parallel bodies may be found elsewhere, e.g. in [18]).

Let $w_i : K \to K$ for $i = 1,2,\ldots,N$ be a collection of contraction mappings with corresponding ratios of contraction $0 \le s_i < 1$. Then $\{K, w_i : i = 1,2,\ldots,N\}$ is called a contractive iterated function system [5].

Lemma 1: Let $\{K, w_i : i = 1,2,\ldots,N\}$ be an IFS, and define $w : H(K) \to H(K)$ by

$$w(A) = \bigcup_{i=1}^{N} w_i(A) = \bigcup_{i=1}^{N} \{w_i(x) : x \in A\} \tag{1.2.2}$$

for $A \in H(K)$. Then w is a contraction mapping with

$$h(w(A), w(B)) \le sh(A,B), \text{ for all } A, B \in H(K), \tag{1.2.3}$$

where $s = \max\{s_i ; 1 \le i \le N\}$.

Proof:

$$\begin{aligned}
&h(w(A), w(B)) \\
&= \sup_{x\in A, j}\inf_{y\in B, j} d\left(w_i(x), w_j(y)\right) + \sup_{x\in B, i}\inf_{y\in A, j} d\left(w_i(x), w_j(y)\right) \\
&\le \sup_{x\in A, i}\inf_{y\in B} d\left(w_i(x), w_i(y)\right) + \sup_{x\in B, i}\inf_{y\in A} d\left(w_i(x), w_i(y)\right) \\
&\le sh(A,B).
\end{aligned}$$

Banach's fixed point theorem states that a contraction mapping on a complete metric space has a unique fixed point, and as a corollary this point is an attractive fixed point [39]. Because h is contractive, it follows that an IFS possesses a unique attractor $A \in H(K)$ defined by

$$A = \lim w^{\circ n}(A) \tag{1.2.4}$$

where $w^{\circ 0}(A) = A$ and $w^{\circ n}(A) = w\left(w^{\circ(n-1)}(A)\right)$ for $n = 1,2,\ldots$, and that $\boldsymbol{A}$ is independent of $A \in H(K)$. When $K \subset \mathbf{R}^m$, it usually occurs that the Hausdorff-Besicovitch dimension of $\boldsymbol{A}$ is non-integer [5], in which case $\boldsymbol{A}$ is most certainly a fractal.

An approximation of the attractor $\boldsymbol{A}$ of an IFS can be computed by computer by adapting the following method of calculation. Define a probability vector $\mathbf{p} = (p_1, p_2, \ldots, p_N)$ with each $p_i > 0$ and $\sum p_i = 1$. Start from $x_0 \in K$ and define a sequence $\{x_n\}$ by choosing successively

$$x_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), \ldots, w_N(x_{n-1})\} \quad (n = 1,2,3,\ldots) \tag{1.2.5}$$

where probability p_i is attached to the choice $x_n = w_i(x_{n-1})$ once x_{n-1} has been chosen. Then

$$A = \{y: \text{there is a subsequence } x_{n_i} \to y\} \tag{1.2.6}$$

Observe that $a \in \boldsymbol{A}$ if and only if each open neighbourhood of a contains infinitely many elements x_n. When $K = [0,1] \times [0,1] \subset \mathbf{R}^2$ for example, such a computation is relatively easily done for a sufficient number of points. A computer would plot, say, $\{x_n : n = 51,52,\ldots,500,000\}$.

An alternative method for the computer rendering of IFSs is that of [51].

1.2.2 Inverse Problem

The inverse problem is that of choosing contraction mappings $w_i : K \to K$ such that their attractor is or approximates a given set. For example, given a target set L and an $\varepsilon > 0$ the problem is to find an IFS with attractor A such that

$h(A,L) < \varepsilon$. Barnsley *et al* [6] suggest how this problem may be solved in certain cases. Their method relies on the Collage Theorem which, for reasons which should become clear later, might be termed the "Fundamental Theorem of Fractal Image Compression!"

Collage Theorem: Let $\{K, w_i : i = 1,2,\ldots,N\}$ be an IFS. Let a subset L of K be such that

$$h\left(L, \bigcup w_i(L)\right) < \varepsilon \tag{1.2.7}$$

for some $\varepsilon > 0$. Then

$$h(L,A) < \frac{\varepsilon}{1-s}, \tag{1.2.8}$$

where A is the attractor of the IFS.

Proof:

$$\begin{aligned} &h(A,L) \le h(A,w(L)) + h(w(L),L) \le sh(A,L) + h(w(L),L) \\ &\Rightarrow (1-s)h(A,L) \le h(w(L),L) \\ &\Rightarrow h(A,L) \le \frac{h(w(L),L)}{(1-s)} \end{aligned}$$

Hence only a collage, or "lazy tiling," of L need be made, by continuously distorted copies of itself to find a suitable IFS.

Sets defined by the above method can be simple or complex. Figures 1.3 and 1.4 show some examples in $\mathbf{R}^2$.

1.2.3 Summary

The method of approximation suggested by Barnsley *et al* is to continuously alter an IFS - rendering and re-rendering - until the attractor of the IFS is

acceptably close to the set it is to approximate. This is possible as an attractor is continuously dependent on its IFS, that is, a small change in the IFS will produce a small change in the attractor.

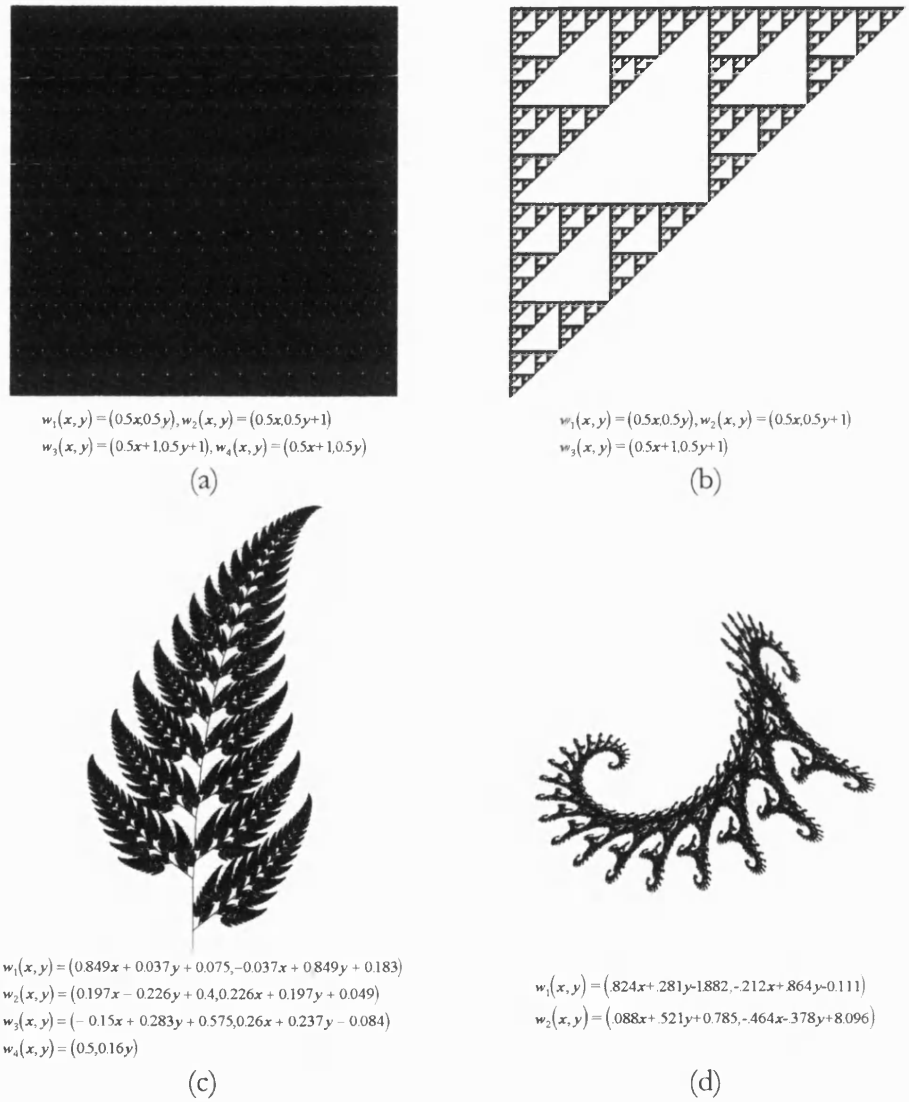
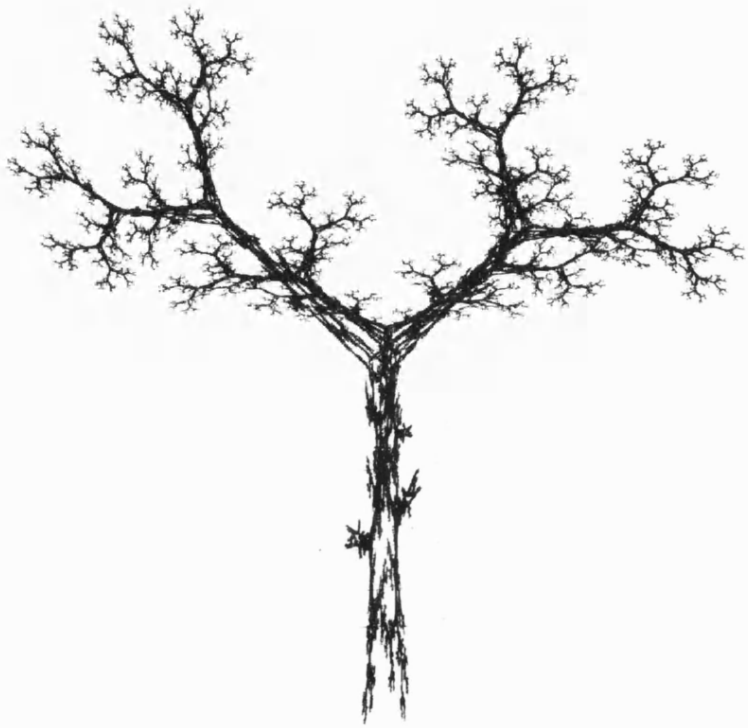


Figure 1.3. Example IFS attractors. (a) Square. (b) Sierpinski triangle. (c) Fern, formula taken from [61]. (d) Dragon, from Fractint 19.5[1]

[1] Public domain software for generating fractal images. Published by Stone Soup Group. http://spanky.triumf.ca/www/fractint/fractint.html.

Although other solutions have been proposed, for example Forte and Vrscay [21], in general the inverse problem is difficult and no computationally effective solution has yet been presented. However the construction of a broad range of fractal sets from simple transformations is impressive for its own merits.



$w_1(x,y) = (0.195x - 0.488y + 0.4431, 0.344x + 0.443y + 0.2452), w_2(x,y) = (0.462x + 0.414y + 0.2511, -0.252x + 0.361y + 0.5692)$
$w_3(x,y) = (-0.058x - 0.07y + 0.5976, 0.453x - 0.111y + 0.0969), w_4(x,y) = (-0.035x + 0.07y + 0.4884, -0.469x - 0.022y + 0.5069)$
$w_5(x,y) = (-0.637x + 0.8562, 0.501y + 0.2513)$

Figure 1.4. Tree. Formula from [61].

1.3 Image Compression

Image data compression is concerned with minimisation of the number of information carrying units used to represent an image [34]. An image which has been transformed into a more efficient representation is termed a "compressed image." In the "lossy" case, of which fractal image compression is an example, a certain degree of deterioration is tolerated to achieve greater compression.

Below some essential aspects of the subject are described: image acquisition, representation of colour, and distortion and compression measurement.

1.3.1 Image Acquisition

Many images which require compression are computer generated, for example graphics used in computer games. However more commonly images are of photographic origin, taken from a real scene using a digital camera or scanned from a photograph. Below, the basics of grey-scale image acquisition are explained - an explanation of colour is beyond the scope of this work.

A camera works by using a lens to focus the light from objects onto the image plane of the camera. The use of a lens causes spherical and chromatic aberration, but the most important feature, at least from the point of view of image compression, is the sampling of the image.

An image produced in a digital camera is usually converted to an electrical signal by the use of Charge Coupled Device (CCD) cells. These are MOS capacitors that accumulate charge proportional to the number of photons that are incident on them [8]. Typically a CCD array samples at 752x582 pixels, which is common mainly because of television standards. The analogue charge in each CCD cell is then converted to a digital value, and the intensity of the signal is split into 2^n grey-levels, where $n = 8$ is common.

A scanner captures images by reflecting off an opaque original or shining light through a transparent original into an array or strip of photosensitive (light) cells (e.g. CCD cells.) The subsequent process is similar to that in a digital camera.

1.3.2 Colour Representation

In image compression colour is usually represented in one of two ways. Firstly by a set of red, green and blue (RGB) 8-bit images, which can then be combined to produce a displayed or printed image in colour. Secondly by a

luminance and two chrominance (YUV[2]) images, a form where distortion in the chrominance (colour) images is much less perceptible and hence is more useful for image compression. The YUV form is a linear transformation of RGB images, however there are many other such transformations.

In image compression development it is normal for only grey-scale images to be considered, as colour images may be approximated in the same way and at higher compression ratios. Additionally to the human visual system the chrominance component of colour images is by far the most important.

1.3.3 Measurement of Compression and Distortion

The amount of compression an image is subjected to is measured in one of two ways. Either by *compression ratio* which is the ratio of the amount of data used to describe the image in its original form to the amount of data used to describe the compressed image, or by *bits-per-pixel* - the compressed data size in bits divided by the number of pixels in the image.

The measurement of distortion is more difficult but is commonly measured with one of the following formulae [34].

Mean Square Error (MSE):

$$MSE = \frac{1}{N}\sum_{i,j}\left(f(i,j)-g(i,j)\right)^2, \tag{1.3.1}$$

Peak Signal to Noise Ratio (PSNR)[3]:

---

[2] For convenience the initials YUV are used to denote the luminance-chrominance images, although in television engineering YUV usually denotes the modulated signals themselves.

[3] Many researchers use the maximum possible image intensity (typically 256) instead of the maximum found in the image. Comparison with their results is possible using the signal peaks given in Chapter 4, Figures 4.5.1-4.

$$PSNR = -10\log_{10}\left(\frac{MSE}{Max|g(i,j)|^2}\right), \tag{1.3.2}$$

Signal to Noise Ratio (SNR):

$$SNR = -10\log_{10}\left(\frac{\sum_{i,j}\left(f(i,j)-g(i,j)\right)^2}{\sum_{i,j}g(i,j)^2}\right), \tag{1.3.3}$$

In the above N is the number of pixels in the image, g is the original image, and f is the compressed image.

Difficulties arise as these methods for measuring image distortion do not take into account how the error is interpreted by the human visual system. Work done on this problem [85] suggests that this is not necessarily serious when comparing compression methods which differ only slightly.

1.4 Compression Methods

1.4.1 Image Compression Methods

Image compression methods vary widely in their working, but a typical method has the following stages

1. The original digital image is transformed into another domain, where the new set of data has the property of greater, ideally optimal, packing of the information content into a smaller number of the coefficients.

2. Some values of the data set are truncated or discarded, or otherwise quantized, in order to save space in the eventual file. The amount of loss depends on the required degree of compression.

3. Statistical redundancy in the data is removed by entropy coding, and the data is then stored or transmitted.

The decompression of the image reverses this process, except of course for step 2 which causes loss of information in the recovered image. Some methods also lose information at Step 1, i.e. the transform is not invertable. Fractal methods to date have been of this type. Often these methods are used because they have the advantage of requiring fewer operations to compute and give good performance at high compression ratios.

1.4.2 JPEG Image Compression

The Joint Photographic Experts (JPEG) defined a standard for image compression in 1988 [78]. This standard is now in very wide use. It is based on the Discrete Cosine Transform (DCT) [2]. The general form of the 2-dimensional DCT is shown in equations 1.4.1 and 1.4.2:

$$g(i,j) = \frac{2}{\sqrt{NM}} \sum_k a(k) \left(\sum_l a(l) \cdot c(k,l) \cos\left(\frac{(j+0.5)l\pi}{M} \right) \right) \cos\left(\frac{(i+0.5)k\pi}{N} \right) \tag{1.4.1}$$

$$c(k,l) = \frac{2}{\sqrt{NM}} \sum_j a(k) \left(\sum_i a(l) \cdot g(i,j) \cos\left(\frac{(j+0.5)l\pi}{M} \right) \right) \cos\left(\frac{(i+0.5)k\pi}{N} \right) \tag{1.4.2}$$

Where N, M are block side lengths, $g(i,j)$ is the image pixel intensity at (i,j), $c(k,l)$ are the transform coefficients at (k,l), and

$$a(l) = 1 \text{ for } l \neq 0 \tag{1.4.3}$$

$$a(l) = \frac{1}{\sqrt{2}} \text{ for } l = 0 .$$

The DCT is an orthogonal, 2-d separable transform and various algorithms exploit these facts, and its partial symmetry, to compute the forward and inverse version [14].

1.4.3 Wavelet Transforms

Research into the use of wavelets for image compression has increased rapidly over recent years, inspired by the theory of discrete wavelet transforms [15,42]. Wavelet coding is based on sub-sampling high- and low-pass filters, matched in such a way that they split the data without losing any information. For image compression a common choice is one of the family of Daubechies orthogonal filters [15] or the bi-orthogonal filters [12,73].

To compress the transform coefficients Shapiro [70] uses a zerotree method which exploits the pyramidal nature of the wavelet filtered image. Said and Pearlman achieved improved results with a "Set Partitioning in Hierarchical Trees" (SPIHT) algorithm which has more efficient significance map coding and produces amongst the best image compression results of any technology to date [67].

1.4.4 Other Methods of Image Compression

A range of other methods exist for image compression. For example the DCT may be used much more efficiently than in the JPEG standard. In [48] it is shown how using a 16x16 partition of the image the DCT coefficients may be organised into a pyramid structure and the zerotree algorithm of Shapiro applied, with very good results.

Another popular method is vector quantization where image blocks are approximated as linear combinations of blocks, or vectors, from a fixed 'library' of vectors. The key problem here is in the design of the library, since its size greatly effects computational requirements.

1.4.5 JPEG 2000

A successor to the JPEG standard, based on wavelet compression, is now under development by ISO/IEC. The main objective of this standard is to achieve low bit-rate image coding with rate-distortion and subjective image quality performance superior to existing standards. The standard should be

completed by the end of the millennium and is intended to offer state-of-the-art compression for at least ten years. The contribution phase of JPEG 2000 consisted of a request for proposed methods and a performance test of the technology using 35 test images and has now been completed, with 25 proposals in total.

In the next stage of JPEG 2000, the convergence phase, best compression technologies are to be integrated to yield the best possible compression algorithm.

1.5 Quantization and Entropy Coding

1.5.1 Quantization

The transformation of an image from its original form produces a large set of coefficients. The coefficients may take a very large range of values, limited only by the means of storing them in computer memory. Quantization takes these coefficients and produces integer values, for example in the range 0-255. This process dramatically reduces the storage cost of the coefficients but also introduces error. The more 'coarsely' the coefficients are quantized the less space they require, but the more inaccurate they are compared to the original.

Uniform quantization is the most basic form of quantization and will be used throughout in this work, as it is an efficient method and reduces the number of variables when designing compression schemes.

To quantize a coefficient it is divided by a quantization factor and rounded to the nearest integer.

$$c' = Round\left(\frac{c}{Q}\right) \tag{1.5.1}$$

The quantized coefficient therefore has a maximum value, which depends upon the maxmimum value of c.

To reconstruct the coefficient value

$$c = c' \cdot Q \tag{1.5.2}$$

Some sets of coefficients may be more important than others, for example the DC or grey-level coefficients of the DCT, so it may be necessary to choose different values of Q for each set of coefficients.

Other types of quantization exist, such as Lloyd-Max quantization [45], however uniform quantization has the added bonus of being scaleable - when the quantizer is applied to image blocks of different sizes the quantizer has the same effect, simplifying the overall problem. Also it has been shown that often entropy coding following uniform quantization is as compact as Lloyd-Max [71].

1.5.2 Entropy Coding

Once quantized coefficients have been created they are coded losslessly to minimise the amount of space required for storage or transmission. This is done by exploiting the statistical properties of the data set.

In lossless coding it is useful to refer to the input as data symbols to be compressed and the output from the lossless coder as compressed symbols. The data symbols are usually quantised transform coefficients in image compression, but they can be anything, provided the coder has a knowledge of their statistics.

The entropy of a data set is the average minimum number of bits that a data symbol stream can be compressed into:

$$Entropy = -\sum_{x} \Pr(x) \log_2(\Pr(x)) \tag{1.5.3}$$

where $\Pr(x)$ is the probability of x occurring.

The two most common methods of entropy coding are Huffman coding [28] and arithmetic coding [55]. Huffman coding is easier to implement and is thus used in this work.

1.5.3 Huffman Coding

Huffman compression [28] is designed to represent a data set in a form requiring close to the theoretical minimum number of bits required, given by equation 1.5.3. This is achieved by representing common data symbols with shorter symbols and rare data symbols with longer symbols. The average effect of this method is to reduce the number of bits required overall. When the symbol probabilities are negative powers of 2, optimal coding efficiency can be achieved.

The Huffman coder forms a tree from the data using the probability of each symbol. The rules for its construction are given below.

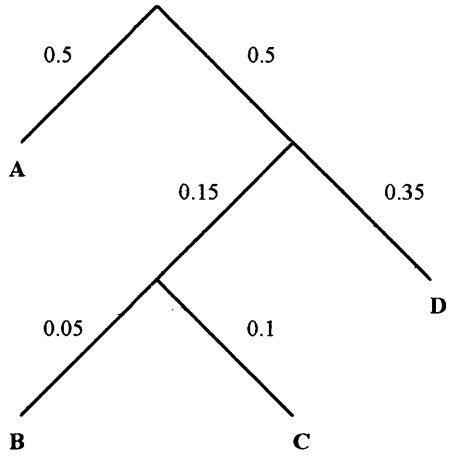
- First consider each data symbol x as a one node tree with probability $\Pr(x)$.
- Link the two trees with lowest probabilities and recalculate the new tree's probability
- Continue to link trees until only one, complete, tree remains.

Figure 1.5 shows a sample Huffman tree and table for symbols A, B, C, D with probabilities 0.5, 0.05, 0.1, and 0.35 respectively.

To encode a data set with Huffman coding a tree structure is created using the recursive algorithm described above. From this tree a *translation table* is generated, giving a variable length Huffman code for each source symbol. The table is created by traversing every path of the tree and storing a zero when a left branch is chosen and a one when a right branch is chosen. At the end of each path is a source symbol and the sequence of ones and zeros that led to

this symbol is the Huffman code. Any source symbol may now be encoded using this table.

•



Symbol	Code
A	0
B	100
C	101
D	11

Figure 1.5. Huffman tree and corresponding table.

To decode a symbol from a stream or string of bits (i.e. a sequence of bits read from a file), one bit is read at a time and the Huffman tree is traversed, choosing a left or right branch at each node depending on the value of the bit. Eventually a leaf of the Huffman tree is reached and this gives the symbol value. This process may be carried out efficiently using a table with three columns. The first column contains a numbered list of nodes and leafs. For each node the second and third columns are the node or leaf numbers corresponding to choosing a left or right branch. For each leaf the second column contains the decoded symbol, and the third column is left blank.

1.5.4 Other Methods

Other methods may reduce the entropy level of the data itself. For example delta coding creates a data set where the difference between symbols is stored rather than the symbols themselves. This is useful if there is a simple correlation between adjacent symbols.

Another useful method is run length coding where 'runs' of the same symbol are replaced by a pair of values - the symbol itself and the number of occurrences.

1.6 Fractal Image Compression

The principle of fractal image coding is that the image can be expressed as the attractor of a transform acting in the space of images. This idea stems from the work of Barnsley *et al* on IFSs, where complex shapes are produced from using simple transforms, and will be explained in detail in later chapters. The work of Jacquin [35] is the first published fractal image coding system, and the next chapter begins by reviewing Jacquin's method and the developments from it.

Chapter 2

LITERATURE REVIEW

The vast majority of published work on fractal image compression derives from the method of Jacquin [35,36]. The method works by partitioning an image into square blocks called *range* blocks and determining the fractal code for the image block-by-block. Each block is encoded with two terms: a *massic* term representing the grey-level of the block, and a *geometric* term, representing its similarity to a *domain* block elsewhere. By this method the self-similarity of the image is captured.

The result is an image transformation which when applied iteratively (see section 1.2) converges to an approximation to the original image. Research has been carried out into each aspect of Jacquin's algorithm, primarily block transform design, quantization, partitioning, searching, and decoding. Each of these will be treated below.

2.1 Jacquin's Method

2.1.1 Encoding

The support S of a digital image g is partitioned into non-overlapping square range cells of two different sizes. The larger squares are of size $B \times B$ and the smaller ones of size $B/2 \times B/2$. In the original work these are termed *range parent* and *range child* blocks respectively, however there are differing uses of these words in the literature and the term *parent* is reserved here for another purpose and the terms *range* and *child* are used interchangeably.

Larger range blocks are used to code smoother areas and smaller blocks to capture detail in more complex areas, using the *root-mean-square* distance to measure distortion between the image g and its approximation f:

$$d(f,g)=\sqrt{\sum_{i,j}\left(f(i,j)-g(i,j)\right)^2} \tag{2.1}$$

A pool of domain blocks is defined using the original image. These blocks are twice the size of the range blocks and are found by sliding a window across the image, moving in steps of one or more pixels. The step size is typically chosen as $B/2$.

The domain blocks are then classified based on their geometric features as either *shade blocks*, *edge blocks*, or *midrange blocks*. A shade block is "smooth" with no significant gradient, an edge block has a strong change of intensity, and a midrange block has moderate gradient but no significant edge. The edge blocks are further split into *simple edge* blocks and *mixed edge* blocks. The shade blocks are removed from the pool as they remain shade blocks under every possible block transformation and so can only be used to approximate other shade blocks. Such blocks, however, may also be approximated by the massic part of the transformation [36].

Each range block is classified and coded. If the block is a shade block it is approximated by a uniformly grey block. If it is midrange every element of the midrange domain pool is scanned, and the block is approximated by

$$g(x,y)\approx \alpha g\left(w^{-1}(x,y)\right)+\Delta \tag{2.2}$$

where $w(x,y)$ represents the mapping from domain to range, α is a contrast scaling taking one of four values less than one, and Δ is a luminance shift.

If it is an edge block a segmentation of range and possible domain blocks occurs. Each block is assumed to be of sufficiently small dimension to be

segmented into two regions, one *dark*, one *bright*. The dynamic range of a segmented block is then the grey level difference between the bright and dark regions. The range block is then approximated by

$$g(x,y) \approx \iota\left(\alpha g\left(w^{-1}(x,y)\right) + \Delta\right) \tag{2.3}$$

where ι is an isometric mapping of the block. In this case α is computed so that the dynamic ranges of the range and scaled domain block are equal. Of all the edge domain blocks the block which minimises distortion is selected.

2.1.2 Reconstruction

The fractal code may be seen to be a collection of block mappings using a given image whose union approximates the original image. The decoding scheme simply consists of iterating the set of block mappings, or *fractal mapping*, until convergence to a stable decoded image is observed. Results in [36] suggest around 10 iterations are required for convergence.

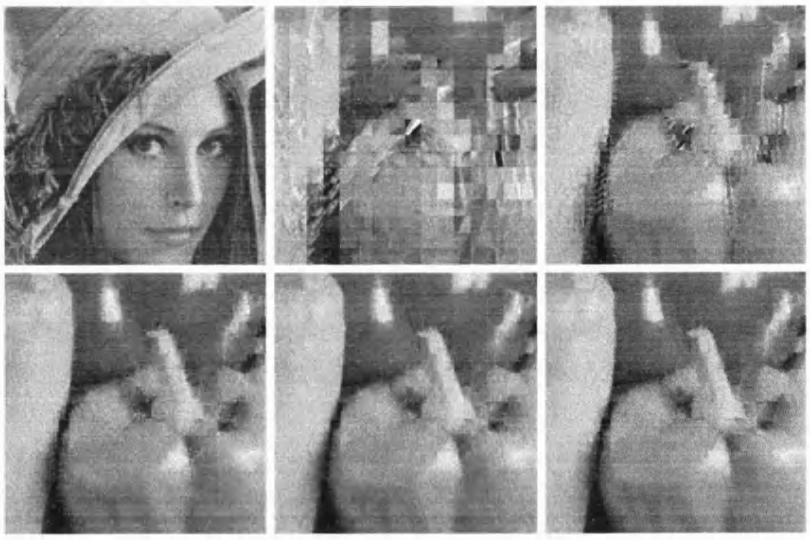


Figure 2.1. Encoding of "Peppers" image, decoded using "Lena" as starting point.

2.2 Transform

The block transform of Jacquin's scheme has the greatest impact on pixel values within a range block, and alterations here have a direct impact on the performance of the codec. A range of enhancements have been proposed.

Firstly, almost all fractal coding schemes use the 2:1 spatial contraction of domain blocks onto range blocks, but in fact this is not necessary for convergence [20]. Increasing the contraction to 3:1 has been found to improve decoder convergence [11] but there appear to be few theoretical or practical results for any one factor. Gharavi-Alkhansari and Huang [23] use a combination of 2:1 and 1:1 domain sizes. Øien and Lepsøy [58] demonstrate that the domain-to-range decimation affects the accuracy of the collage error prediction, with 4:1 decimation producing the most accurate prediction of the approximation error.

In Jacquin [35], decimation of the domain blocks to the size of the child blocks is achieved by the averaging of groups of pixels, but other methods may be used. In [7] it is reported that the use of an anti-aliasing filter gives superior results, whereas in [41] the computational cost is reduced slightly by taking one of every four pixels and discarding the rest.

The symmetry operations utilised by Jacquin are widely used. Although Jacquin found some of the isometry operations were used more than others, other researchers found the probability distribution of their usage was relatively flat [82]. In spite of their widespread usage, there is growing evidence that their application is counterproductive in a rate distortion sense [82, 69].

A range of block shapes can be used, corresponding to different ways of partitioning the image. Many polygonal shapes have been tried; particularly common are rectangular block methods in horizontal-vertical decomposition of the image, and triangular blocks in triangular partitions. Section 2.4 deals with this subject in more detail.

To attempt to improve the quality of the geometric part of the transform a method has been proposed in [74] called *orthogonal basis IFS*. Determining a basis from the domain blocks in the image, each range block is approximated by a linear combination of domain blocks, with the most effective blocks used. This method is complex, paticularly because the basis is determined using a relatively complex method. However the method has potential since much better approximations can be produced to each block.

[23] presents a similar but slightly more general method, using domain blocks near the range block and a combination of fixed basis blocks such as used with the Bath Fractal Transform, described later. In addition the domain blocks can be either larger as in Jacquin or the same size as the range.

One significant early step was the realisation that the scaling term could be found directly using an inner product space technique [59], and Fisher [19,ch.3] shows that allowing the maximum magnitude of the scaling coefficient to be as high as four gives an improvement in PSNR, although [31] contradicts this for coarse quantization cases. In addition to the above proposals attention has focused on improvement of the computation of the transform coefficients.

The technique of mapping the original image to itself to find the solution derives from the collage theorem (section 1.2.2). However the collage theorem only gives an upper bound on the approximation error.

In [58,57] it is demonstrated that a better bound may be found by using a slightly modified transform, where the DC level of domain blocks is removed, and in the same works a method of direct optimisation is presented. However it is not proven that these methods lead to significant gain in the rate-distortion sense.

In [29] the fractal code is determined by exhaustive search to see to what degree collage error coding is suboptimal. The result is an average 1.5 dB

improvement in SNR with a fixed block size partition - demonstrating that this area of investigation is of significant importance.

The grey-level removal from domain blocks, suggested by Øien *et al* [58] is referred to as the *orthogonal transform*, and is shown to converge more rapidly than the original form. Part of the contribution of this thesis is the application of an othogonalisation operator to the Bath Fractal Transform.

An alternative extension to the transform proposes manipulation of the domain block in the frequency domain [7]. The spectral coefficients are split into a small number of regions and in each region the coefficients are scaled by different factors. The result is a transform which approximates details in blocks better than Jacquin's method and converges more quickly. The overall subjective quality at a given bit rate is described as improved.

Finally the transform can be improved by altering the massic part of the transform. The most well-known case is the Bath Fractal Transform (BFT) [47] which is explained in detail later, and forms the basis for further work. In essence the grey-level term is added to by a number of bi-polynomial terms. This improves the approximation quality although increases the bit rate. Overall, it has been shown to be more efficient than Jacquin's method.

2.3 Quantization and Entropy Coding

Quantization and, optionally, entropy coding are a key to achieving compression through fractal approximation and several researchers have tackled this problem.

[19] contains a comparison of the bits used to quantize the scaling factors, between three and seven bits, and grey level, between five and nine bits, using linear quantization at a range of compression factors. Different allocations performed better at different compression rates, but Fisher concludes that five bits is the best overall choice, which is higher than Jacquin's original six or

seven bits to quantize the grey level uniformly and two or three for the scaling factor [35]. [19] also advises that Huffman or arithmetic coding should subsequently be used.

[31] examines the scaling coefficient and offset for any correlation and discovered a correlation of 0.7 to 0.95. This relationship may be exploited to reduce the bit requirement of the scaling coefficient, but an alternative is the use of the orthogonal transform of Øien *et al* [60].

[72] draws the conclusion from geometrical analysis that higher coefficient values should in fact be quantized more finely, since their impact is greater. Uniform quantization, it is argued, leads to the over representation of simple blocks and weakened performance on highly textured blocks.

Whilst uniform quantization is the most common method, the probability density functions of scaling and grey scale coefficients [36] suggest either non-linear quantization or the additional use of entropy coding, typically Huffman or arithmetic coding.

The domain block locations and isometries are stored without further compression in [35] and this appears to be the norm, with little to suggest a better method. Storage of domain block locations is most commonly done by storing offset parameters or using an index related to the searching method used (see later).

A more advanced aspect of the problem is the distribution of bits efficiently throughout the image. An example is [77] which determines the effect of using a scaling coefficient in a block and compares its rate-distortion effect with that of splitting a high error block. The coefficient is used only if it is the more efficient use of bits.

2.4 Partitioning

Jacquin's partition of the image consisted of a simple two-level quadtree. Many other partition types exist and have been used successfully in other image compression schemes. The main ones: quadtree, horizontal vertical (HV), triangular, and polynomial have been tried with fractal transforms.

2.4.1 Fixed block sizes

Fixed block size partitions are the simplest possible type, and have received little attention because the block transform itself cannot account for varying amounts of detail in the image. It is logical therefore to use partitions which vary in block size and can use small blocks for high detail areas and large blocks for low detail areas.

The exception to this rule is the Bath Fractal Transform which uses a multiple grey scale terms to achieve better rate-distortion results. In [53,83] the quantization was varied to give different compression ratios with the same block partition. The conclusion of [83] however is the same as above, that optimal performance can only be achieved with an adaptive partition.

2.4.2 Quadtree

The simplest and often most effective adaptive partition is the quadtree, where large blocks from a simple initial partition are recursively split into sets of four smaller blocks, using the approximation error as the criterion for splitting. The result is a tree structure in which every non-terminal node has four descendants, and a partition of the image where the approximation error is distributed quite evenly throughout. A simple partition is illustrated in Figure 2.2.

An improvement to the quadtree scheme for fractal image compression was proposed by Reusens [63]. Quadtree blocks were overlapped to eliminate block artefact and the result was a decrease in the artefact with no increase in bitrate.

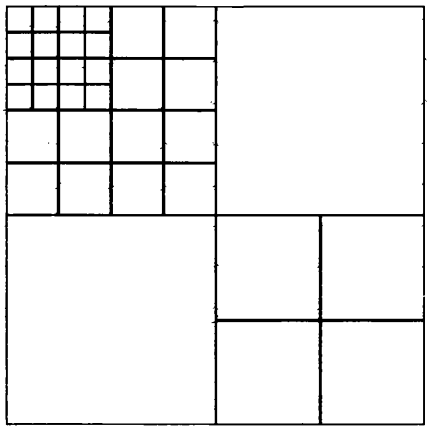


Figure 2.2. Example quadtree partition.

2.4.3 Horizontal-Vertical

A HV scheme works in a similar way to a quadtree but divides rectangular blocks with high error into two smaller rectangles. The line of division may be found by exhaustive search or based on analysis of the blocks geometry. For example blocks can be split along strong vertical or horizontal edges [19, Chapter 6]. Figure 2.3 illustrates a simple HV partition.

Clearly HV partitioning is more flexible than the quadtree method, but it is also more complex and requires a higher proportion of the total bit allowance to describe it. In addition domain searching is more complicated, and some compensation must be made for the varying aspect ratios of the range blocks. However the comparisons of Fisher [19] show HV partitions perform significantly better for most of the compression range which is of interest. At high compression rates of around 40:1 and above the quadtree is superior for the Lena image but otherwise the HV partition is clearly superior.

2.4.4 Triangular

Triangular partitioning has also been investigated [56] and shows good visual quality compared to the quadtree. Blocking artefact is not as obvious, however the extra computational cost and the complexity of the partition meant this

partition has not been widely used. Figure 2.4 shows a simple triangular partition example.

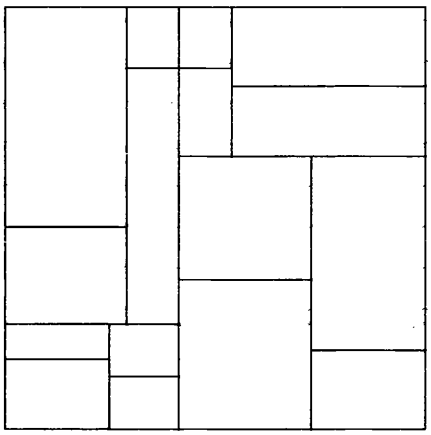


Figure 2.3. Simple HV partition.

2.4.5 Polygonal

Reusens [64] compares the above quadtree and HV partition types and experiments with a polygonal partition where the image is divided into polygonal regions having various numbers of sides. It is stated that the quadtree partition accounts for 7% of the total bit cost, the HV partition 20%, and the polygonal partition 25%. The rate distortion results for Lena show that the greater the proportion of bits allocated to the transform coefficients, as opposed to the tree structure, the better the algorithm performs.

2.4.6 Summary

To conclude, quadtree partitioning is a simple and fast method, but allows only adaptation in size, not shape, to the characteristics of the image. Other methods are reported to give better visual results, but it is not clear which is the best method overall. The results of Reusens and Fisher are contradictory, but there are a number of differences between their implementations of the two methods which may account for the discrepancy. It seems clear however that either a quadtree or HV partition will be the most effective option.

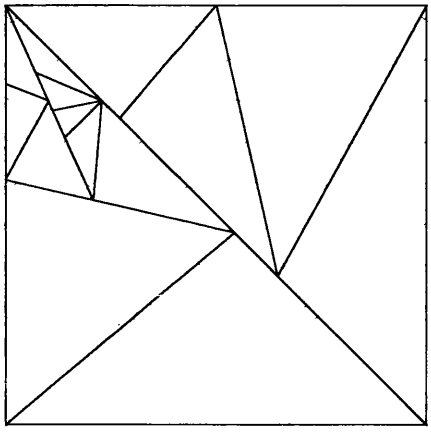


Figure 2.4. Example triangular partition.

2.5 Searching

The method of searching for domain-range matches is a topic containing a plethora of ideas. It is the very essence of Jacquin's method, but is remarkably slow, requiring a large pool of blocks to be tested for each range block to be coded.

One method of acceleration is the orthogonal transform presented in [58]. The candidate domain blocks are orthogonalised with respect to the grey-scale mapping and consequently only the optimal scaling coefficient need be computed when each new domain block is considered. As a result the cost of evaluating domain-range block matches is reduced. Additionally the use of quadratic polynomial terms in the grey-scale mapping can allow many more blocks to be classified as smooth, further reducing the computational cost of the encoding.

Commonly though, accelerated searching schemes work by classifying domain blocks so that only similar blocks are tested for a match against the range during the matching step. Many classification schemes are known and often combinations are possible.

As explained earlier, Jacquin used a scheme that created four subpools of domain blocks: shade blocks, simple and mixed edge blocks and midrange blocks. For each range block, of one of these types, all domain blocks were tested from the corresponding pool. However further classification or characterisation of blocks in each subpool has been found to be effective. By quantifying in advance some characteristic of the domain blocks, when a range block is to be encoded only those domain blocks which have similar characteristics need be searched.

Numerous techniques exist for this purpose, for example:

- Frigaard *et al* [22] propose a *feature space* where each domain block is represented by an n-dimensional vector with each co-ordinate continuous, and representing a characteristic of the block. The characteristics used were the grey-level standard deviation of the block and the number of dominant grey levels.

- Boss and Jacobs [10] used a system of *archetypes*, where there were many classes of range block each represented by an archetypal member. The complex process of determining the archetypes proved somewhat counter productive however.

- Fisher [19,Ch3] chose three domain pools, based on block sizes, and used a classification scheme to reduce the complexity of the domain-range matching step. The classification used the mean values and variances of quadrants of blocks to produce 72 classes.

- Barthel and Voyé [7] used an ordered search which spiraled out from the range block and stopped if the approximation error fell below a threshold.

- Hürtgen and Stiller [32] combined a hierarchical search algorithm with block analysis giving search time which is only a fraction of Jacquin's method and gives similar fidelity.

- Saupe [68] used a method based on nearest neighbour searching, in a high dimensional space.

Fisher [19,Ch.3] examined the theory that local self-similarity is more common, but concluded that the increased likelihood of finding a match close by is not caused by local self-similarity but instead by there being more domain blocks to choose from close to any given range. Although this conclusion is at odds with that of Barthel and Voyé [7].

Conclusion

Domain pool searching is of only indirect relevance to this work, and consequently the above is only a summary of some of the ideas in an area which constitutes a large proportion of the fractal image compression literature. It is not clear, even to specialists in this area, which method is most effective, but this remains a fascinating area of intensive research and investigation.

2.6 Decoding

The approximation to the image is found using the fractal code by iterative application to any initial image, as illustrated by Figure 2.1. Most fractal mappings are linear [30] and so may be written in the form $h \to Th + b$. Evaluating the expression

$$f = (I - T)^{-1} + b \tag{2.4}$$

therefore gives the attractor image f, but is computationally more costly than the standard iterative decoding scheme used by Jacquin [35].

Convergence

The main problem at the decoding stage is the convergence of the iteration sequence. Restricting the magnitude of the scaling factor (2.2) and (2.3) to less than 1 in magnitude guarantees convergence. However in [19] and elsewhere it is noted that the magnitude limit can be raised to increase performance. This leads to problems as there is no obvious guarantee of convergence, and for a useful implementation of a fractal coding scheme reliability in this sense will be required. Other conditions for contractivity have consequently been studied.

For example, in [30] it is shown that the convergence properties of various coding schemes can be analysed in a statistical sense. The fractal mapping can be split into one or more mapping cycles, each of which must converge for the full mapping to converge. If the spectral radius of a cycle is less than one it will converge [30].

As a result the probability density functions of the eigenvalues for various choices of design parameters are derived, and the probability of a divergent coding is computed. The results suggest a large domain to range size ratio and longer mapping cycles are more likely to produce convergent fractal codes.

Similarly Kominek [37] observes that the contractivity varies between pixels and is the product of all scalings under which the pixel is mapped. The term *eventual contractivity* is introduced for mappings which form a contraction when applied several times. This, it is noted, is a sufficient condition for convergence of the transform as a whole.

Kominek [37] also introduces the term *partial contractivity* which is useful when cycles are long but only a limited number of, say four, iterations are applied. Only the first part of the cycle will have an effect, and may not be contractive. In this case 'speckle noise' may be seen in the image.

Fast Decoding

The efficiency of iterative decoding has also been examined. Whilst fast compared with the encoding stage, there is still opportunity for improvement, primarily through the use of pyramid decoding schemes [3,50].

Baharav *et al* [3], for instance, exploit the continuous basis of IFS theory to prove that fractal codings can be decoded using a pyramid scheme, and find that the decoding can be accelerated by an order of magnitude. [50] presents an equivalent scheme, specific to the BFT, which has its origins in the fast rendering algorithm of [51].

Zooming

Another aspect of the fractal decoding process is the possibility of increasing the image resolution or zooming on only part of the image. The fractal transform is independent of resolution by nature so it is simply a question of applying the transform using block dimensions, in terms of pixels, which have been scaled by the required factor.

Fisher [19] presents an increased resolution image first compressed with Jacquin's algorithm, and finds that details are indeed produced at the new scales and that the result is better than the same decoding with pixel replication.

Götting, Ibenthal, and Grigat [26] contains a more thorough study, using up to six polynomial terms for the massic part of the transform. Götting *et al* conclude that the subjective quality of the resulting images is proportional to the order of the polynomial part of the transform, and is inversely proportional to the quantization of the polynomial parameters. Using a 4x4 block size, their x8 magnification of Lenna's eye is superior to conventional nonlinear interpolation.

Polidori and Dugelay [62] identifies the blocking artefact of zooming as a problem area and attempts to compensate with a series of techniques which

overlap blocks. The result is a reduction in blocking effects, seen in [26], and a sharpness of edges, though much distortion even with small block sizes.

Conclusion

Whilst the above discussion has concentrated on results for the classical algorithm, the use of the orthogonal transform allows faster searching at the coding stage and has the added advantage of decoding in a finite number of steps [58]. The use of post-processing techniques for fractal image compression has not been investigated to a great extent, although in [19] a simple averaging technique applied at range block edges produces an improvement in visual quality.

Further work on the challenge of zooming is presented later in this thesis. The above work forms the basis for that study.

2.7 Bath Fractal Transform

This thesis aims to add to the study of the Bath Fractal Transform (BFT) and build upon work already done in its development. For this reason it receives special attention here.

Origins

The origins of the BFT lie in [49], which presents a computationally efficient technique for approximating blocks using a self-affine system, relying on a theory of invariant functions. To explain the workings of the method it is easiest to interpret it using a current fractal image compression framework.

The image is partitioned into domain or *parent* blocks of size 8x8 pixels and each domain block is subdivided into four equal-sized range or *child* blocks. As each child block is contained by its parent block, this parent is used, subject to isometric mapping, *without any searching*, for the geometric part of the transforms.

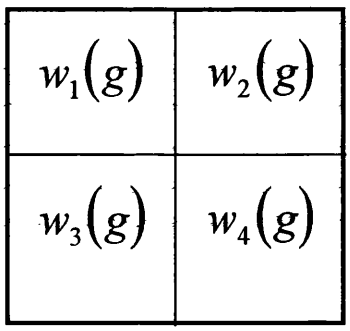


Figure 2.5. Image block g tiled by four children

For the massic part linear polynomial terms were introduced giving an approximation of the form

$$g(x,y) \approx \alpha g\left(w^{-1}(x,y)\right) + \beta + \gamma x + \delta y \tag{2.5}$$

for each block.

The parameters were determined from the collage theorem and were quantized and stored indirectly by storing the function values at corner points of blocks.

The result of this transform design is a method which uses self-similarity to approximate image blocks independently of each other. The elimination of searching from the encoding step makes the method very fast, with symmetrical encoding and decoding times. The independence of the approximation from one parent to another means each block can be decoded independently and this allows simpler decoding algorithms and makes it practical to determining the approximation error during the encoding stage.

Generalisation to the BFT

Monro generalised the above method in [46,47] to include any other bi-polynomial massic expressions and combined the method with localised searching to give a class of transforms of which Jacquin's method was one particular member.

Results presented for a 4x4 child block size over the Gold Hill test image compare the effects on performance of different transform options and show that the image fidelity improves with both the use of searching and the increased complexity of the massic part.

Non-Searching Parameter Determination

Further work on the BFT has sought to clarify design choices to produce the best overall fractal compression method.

By comparing differing degrees of quantization for different blocks sizes it was shown that light quantization with varying block sizes was the most efficient for achieving rate-distortion performance [81].

When comparing the best rate-distortion curves which could be achieved with each order it became clear that the bi-quadratic case was optimal for image compression [53,81].

Geometric Options

Various comparisons of the geometric part of the transform were carried out, and in [54] it is shown that the use of isometries is counter productive in the rate-distortion sense.

The degree of searching to be used was also investigated by Woolley [81]. Varying the distance domain blocks could be from the range block when searching, showed that allowing larger distances gave better fidelity but higher bit cost and that the overall effect on rate-distortion performance was negative. The optimal case was concluded to be that with no searching at all.

In an attempt to improve the performance of the non-searching case, investigations of various divisions of the parent into children were carried out [17,81]. This revealed that allowing the split point to vary yields better rate-distortion results, although this result is effectively superseded by the work described below on partitioning.

Partitioning

Based on the success of varying block sizes with light quantization, consideration was given to the use of quadtree and HV partitioning schemes. Comparing collage error, Sobel edge value, and attractor error for the splitting criterion showed that the use of actual error gave only slightly better results than the edge value, whereas the collage error was inferior by a significant amount [81].

A comparison of the HV and quadtree schemes showed quadtree to be the better performer overall.

Conclusion

Most of this thesis is concerned with either improving the BFT, or the introduction of new transforms which rely on BFT work.

2.8 Other Methods

In addition to the previously mentioned improvements to Jacquin's original algorithm, much promising work has been done on hybridising fractal compression with other methods or interpreting fractal compression in other frameworks.

Fractal Wavelet Coding

The most successful of these are the various subband coding schemes [66]. The principle is to decompose the image into a set of subbands and to predict coefficients in one subband from those in higher subbands. In [65] this is done, using a vector quantizer when the domain-range matching procedure fails.

A number of researchers appear to have discovered the possibility of fractal-wavelet coding simultaneously [16,38,79]. In [38] it is shown that the orthogonal transform can be effected in the wavelet domain using a Haar wavelet. An advantage of such schemes is, of course, that decoding takes only a

finite number of steps and convergence criteria are not required. As stated in [38], using other wavelets than Haar leads to a reduction in blockiness.

Davis [16] reports that allowing range blocks which are poorly matched to be stored by their wavelet coefficients allows the algorithm to degrade gracefully for images which are not self-similar.

Krupnik, Malah, and Karnin [38] suggest that this technique may be easily extended for increasing image resolution.

Fractal Matching Pursuit

Matching pursuit is a general and flexible method for solving optimization problems in a range of subjects. It works by using an over-complete set of functions for coding and then determining which functions are most important. Gharavi-Alkhansari and Huang [24] presents an adaptation of matching pursuit to the fractal optimization process. The result is a method which is reported as competitive with the best fractal coding schemes and comparable in performance to wavelet coding.

Conclusion

The work of Culik and Kari [13] proposes the use of weighted finite automata (WFA) for fractal image compression, though this is not summarised here, and other methods are possible. Work on the implicit fractal transform, contained in this thesis, suggests that fractal coding can be hybridised with practically any other method. The above work with subband coding, wavelets, and matching pursuits indicates the potential of this area for further investigation.

2.9 Summary and Conclusions

This survey has summarised the research work done on fractal image compression since its inception.

Jacquin's original method represented a breakthrough for fractal image compression, and much work followed its publication. Most sought to improve and extend the method, but more recently hybrids with other image compression techniques have been introduced.

In parallel to the direct practical development of the work is the increase in understanding of the underlying principles of fractal transforms resulting from work such as that of Forte and Vrscay [13,75]. These publications aim to provide a unifying mathematical treatment of different types of fractal transforms, specifying a set of rules for their construction.

Summary

A wide range of ideas have been published for improving the determination of scaling and massic coefficients, increasing the number of geometric and massic terms, improving the subsampling of domain blocks to range block size, and determining the best domain to range contraction factor. Studies have attempted to determine the optimal quantization and entropy coding methods and the best partition scheme (quadtree, HV, triangular, or polygonal).

The searching part of Jacquin's algorithm has received special attention. Being slow to encode, a wide range of schemes for accelerating the process using classification, hierarchical searching, and fast domain-range match computation have been proposed and investigated.

Concerning the decoding stage, pyramid decoding schemes have been proposed, and analysis of alternative criteria for convergence has been carried out. Also the effects of zooming and increasing image resolution with fractal transforms have been examined.

The BFT added to the body of fractal image compression research and much work has been done to find the optimal design parameters. The degree of searching, number of massic parameters, use of isometries, and partitioning

options have been investigated and the best case has been shown to be that of the non-searching, bi-quadratic case, without isometries, using a quadtree partition with only light quantization of the coefficients.

Hybrid methods for fractal image compression have received attention as well. Recently fractal-wavelet compression has become popular and results from this subject show great promise.

Conclusions

To conclude, fractal image compression is and remains an area of great research interest. To date several algorithms have achieved rate-distortion performance which is close to that of wavelet compression, which is widely acknowledged to be superior at this time.

The design of fractal transforms presents a number of problems, as pointed out by Wohlberg [80], there are few theoretical results on which design decisions can be based and the categories of choices are not independent, in the sense that any comparative analysis of coding performance between different options in one of these categories is usually contingent on the corresponding choices in the other categories.

The answer seems to be to adopt the approach of Monro and Woolley, and others, producing rate-distortion curves for a set of images, varying one parameter at a time to see which value gives the best result. This method should certainly give a good result, though there is no guarantee the optimal solution will be found.

An important area of research which has received little attention is the nature of self-similarity, though this has been examined [80]. The assumption of piece-wise self-similarity is made in every fractal coding scheme, but it is important to understand what degree of self-similarity to expect. After studying this problem, Wohlberg has concluded that a small but significant degree of local

self-similarity is to be expected, but questions whether this implies that fractal image coding is worthwhile.

In the original work presented in this thesis, the development of block transforms for fractal coding is continued. This approach has been shown to have the potential to achieve excellent performance, and justifies further work. The BFT is a development of fractal block coding and the work presented later will significantly increase its effectiveness and extends its performance by taking advantage of local self-similarity to a greater degree.

Chapter 3

THEORY OF FRACTAL IMAGE COMPRESSION

This chapter describes the theory of block based fractal image compression. A mathematical framework is introduced which is designed to describe later work in this thesis.

The framework uses continuous functional representation of images and fractal mappings. Translation of the theory into a discrete setting for practical application is straightforward.

3.1 Introduction

Definitions

Every image has an *image support*, S, which is a rectangular subset of the plane.

$$S = [x_1, y_1] \times [x_2, y_2] \subset \mathbf{R}^2 \tag{3.1.1}$$

An *image* is therefore a single valued function, with domain S.

If the class of images on an image support S is denoted by $\mathbf{X}$, then fractal image compression encodes an image by finding a *fractal image operator* - a mapping $V: \mathbf{X} \to \mathbf{X}$ - which is contractive with respect to some metric d, on the complete space $\mathbf{X}$. I.e.

$$\exists s \in [0,1[\text{ such that } d\left(V(h_1), V(h_2)\right) \leq s d(h_1, h_2) \text{ for all } h_1, h_2 \in \mathbf{X}. \tag{3.1.2}$$

It is known from Banach's *contraction mapping theorem* [39] that a contractive function has a unique *fixed point* which is globally attractive, and consequently when applied repeatedly to any initial point the resulting sequence of points is

convergent to this attractor. I.e. if the fixed point is f and the initial image is h then $V^{(n)}(h) \to f$ as $n \to \infty$, where $V^{(n)}$ is the n-fold composition of V. As V is an operator on the space of images this fixed point is an image.

Collage Theorem

The essence of fractal image compression is the attempt to find a fractal image operator which has a fixed point close to a given (source) image, hence fractal coding has also been termed *attractor image coding.*

The problem of finding V is known as the *inverse problem* (see section 1.2.2) and is generally hard to solve except in specific cases. The partial solution found by Barnsley uses the collage theorem (section 1.2.2), which is redefined for fractal image compression using the terminology introduced in this chapter.

Collage Theorem for Fractal Image Coding. Let V be a fractal image operator, contractive on a metric space $(\mathbf{X},d)$, with attractor image $f \in \mathbf{X}$ and ratio of contraction s. If g is an image in $\mathbf{X}$ then

$$d(f,g) < \frac{d(g,V(g))}{1-s}. \tag{3.1.3}$$

Fractal Coding using the Collage Theorem

When approximating an image g with a fixed point $f \in \mathbf{X}$, the collage theorem gives a bound on the approximation error. Assuming V is described by a number of parameters and is sufficiently simple, an 'optimal' V may be determined by minimising the right hand side of (3.1.3).

Most fractal compression transforms to date have been block-based and the aim of this thesis is to improve their design. For such transforms the collage theorem is easily applied. In each block, V is described by a number of parameters, assumed to be independent of other blocks. As a consequence

parameter choices for individual blocks do not significantly affect the overall ratio of contraction, and the denominator in (3.1.3) is ignored.

Using the RMS metric

$$d(h_1,h_2)=\sqrt{\iint_S \left(h_1(x,y)-h_2(x,y)\right)^2 dxdy} \quad \text{for } h_1,h_2 \in \mathbf{X} \tag{3.1.4}$$

the contribution over an individual block C to the overall error is the square-error of the block:

$$e(h_1(C),h_2(C))=\iint_C \left(h_1(x,y)-h_2(x,y)\right)^2 dxdy \tag{3.1.5}$$

For a class of fractal image operators (3.1.5) produces a problem which can be solved using least squares minimisation.

Whilst work has been done (see Chapter 2) on determining fractal codes by alternative means, the collage theorem method is the most easily used and widely applicable solution to date, and is used in the fractal methods of this thesis.

3.2 Encoding

To code an image, a tiling or partition of the image support S is created, consisting of N cells, denoted by C_k for $k=1,\ldots,N$. A block taken from a cell is termed a *child block*.

For each cell, C_k, the action of V on an image function h supported on C_k is specified by

$$V(h)(x,y)=p_k(x,y)+e_k h\left(w_k^{-1}(x,y)\right),\ (x,y)\in C_k. \tag{3.2.1}$$

Where p_k is a linear combination of n *basis functions*,

$$p_k(x,y) = \sum_{i=1}^{n} c_{k,i} b_i(x,y) \tag{3.2.2}$$

Commonly, as in Jacquin's method, p_k is a constant grey level

$$p_k(x,y) = c_k \tag{3.2.3}$$

though much work on the BFT has used polynomial and cosine functions.

The second term in the right hand side of (3.2.1) expresses the piecewise self-similarity of the child block over C_k to a block elsewhere in the image. The function w_k is a contractive affine map on S, and a block taken from $w_k^{-1}(C_k)$ is called a *parent* or *parent block*.

To take account of the degree of self-similarity a scaling factor, e_k, is applied to the parent. The mapping w_k is usually determined by searching the image to find the parent block which best matches the child.

The scaling factor e_k and the coefficients of the basis approximation are found by minimising

$$e(V(g),g) \tag{3.2.4}$$

over C_k, or equivalently by approximating the target image g over C_k as best as possible by $p_k(x,y) + e_k h(w_k^{-1}(x,y))$.

Once V is known its contractivity may be calculated and checked. If V is not contractive a limit is placed on the maximum magnitude of the scaling factors $\{e_k; k = 1...N\}$ so that $|e_k| < 1$ for all k.

Note: The above method does not generally produce an optimal code for the image even within the capabilities of the fractal method, as only an upper

bound on the approximation error, the collage error, is used to find the transform coefficients. A suitable method of finding the optimal coefficients has yet to be discovered.

3.3 Decoding

To decode the image an image resolution, usually equal to that of the original, is decided. The finite resolution equivalent of V is applied repeatedly to an initial image to form a sequence of iterates. For example if h_0 is the initial image

$$h_{i+1} = V(h_i) \text{ for } i \geq 0 \tag{3.3.1}$$

As explained earlier, the sequence $\langle h_i \rangle_{i \geq 0}$ converges regardless of which initial image is chosen. By measuring $d(h_{i+1}, V(h_i))$ as each new iteration is computed, the decoding procedure is terminated when the distance between iterates falls below a certain threshold.

The *basis image* may be used for h_0.

$$h_0(x,y) = p_k(x,y) \text{ for } (x,y) \in C_k. \tag{3.3.2}$$

This has the effect of accelerating the convergence process, because the basis should be quite close to f.

Decoding at higher resolutions is straightforward, though more iterates are generally needed. A discrete equivalent of V is found as before, with all block sizes increased by the factor of the resolution increase.

If all parent blocks are local to the corresponding child blocks it is possible to 'zoom-in' on parts of an image without decoding the whole image at a higher resolution. In either case the fractal approximation generates additional detail.

3.4 Modifications

3.4.1 Orthogonalisation

In implementations which use searching to find domain-range matches the computational complexity can be reduced by orthogonalising the parent block with respect to the basis [60]. This process effectively removes basis components from the parent. In the case of Jacquin's original transform this means removing the constant grey-level component.

Definition: Two functions are *orthogonal* in a Hilbert space if their inner product is zero.

In this case $\langle h_1, h_2 \rangle = \iint_C h_1(x,y) h_2(x,y) dxdy$ for a cell C.

Any set of functions may be *orthogonalised* using the Gram-Schmidt procedure[1] [39]:

If $h_1, h_2, \ldots, h_m$ are orthogonal functions defined on a cell C, $m \geq 1$, then for any function h

$$\tilde{h} = h - \sum_{i=1}^{m} \frac{\langle h_i, h \rangle h_i}{\|h_i\|^2} \tag{3.4.1}$$

is an orthogonal function, where $\|h\| = \sqrt{\langle h, h \rangle}$.

By using this formula recursively the basis of functions in (3.2.2) may be orthogonalised, making computation of the massic coefficients easier, as shown in [59].

In the same way, the parent block may be orthogonalised with respect to an orthogonal basis. Denoting the orthogonalisation operator by O and the using h to denote the shrunken and mapped parent, defined on a cell C,

$$\tilde{h} = O(h) = h - \sum_{i=1}^{n} \frac{\langle b_i, h \rangle b_i}{\|b_i\|^2}. \tag{3.4.2}$$

Equation (3.2.1) now becomes:

$$V(h)(x,y) = p_k(x,y) + e_k \tilde{h}(x,y). \tag{3.4.3}$$

Lemma: If $\{b_i\}_{i=1}^{n}$ is a set of n orthogonal functions then

$$c_i = \frac{\langle b_i, g \rangle}{\|b_i\|^2} \tag{3.4.4}$$

over each child block.

Proof: By the Projection Theorem [39] there is a unique $r(x,y)$ in the orthogonal complement of the space spanned by the basis, such that

$$g(x,y) = \sum_{i=1}^{n} c_{k,i} b_i(x,y) + r(x,y) \tag{3.4.5}$$

Since $r(x,y)$ is orthogonal to each basis function

$$\left\langle g - \sum_{i=1}^{n} c_{k,i} b_i, b_i \right\rangle = 0 \tag{3.4.6}$$

from which (3.4.4) is easily derived.

3.4.2 Normalisation

For similar reasons of computational efficiency, the basis and parent functions may be normalised. An additional desirable effect is to make quantization easier.

---

[1] The normalisation of functions has been omitted from the procedure and is dealt with later.

If h is a function then the normalised version is simply:

$$h_N = \frac{h}{\|h\|} \tag{3.4.7}$$

Using an orthonormal basis therefore, equation (3.4.4) becomes

$$c_i = \langle b_i, g \rangle . \tag{3.4.8}$$

3.5 Quantization

For quantization of the transform coefficients many methods exist. As explained in section 1.5 and applied in previous BFT work, uniform quantization followed by entropy coding is an efficient method.

For each transform coefficient $c_{k,i}$, $i = 1...n$, and e_k a quantization coarseness factor is defined which represents the width of one quantization bin. This is denoted Q_i for coefficient i and Q_f for the fractal coefficient e_k.

After quantization each coefficient is represented by a quantization bin number, $q_{k,i}$ for $c_{k,i}$, and $q_{k,f}$ for e_k. The reconstructed coefficients are denoted by $c'_{k,i}$ and e'_k.

For a coefficient c with quantization factor Q, the quantization bin is given by

$$q = round\left(\frac{c}{Q} \cdot N\right), \tag{3.5.1}$$

where $round(x)$ is the integer closest to x and N is the side length in pixels of the block[2]. The recovered coefficient c' is therefore given by

[2] The use of the block size in the quantization step causes the quantization error to be spread evenly through the image, as shown later in section 4.2.

$$c' = q \cdot Q \cdot N . \tag{3.5.2}$$

Analysis

The effects of quantization on a discrete image block are studied. Denoting discrete image blocks with capitals, the error formula of (3.1.5) becomes

$$e(A,B) = \|A - B\|^2 = \sum_{i=0}^{N}\sum_{j=0}^{N}\left(A(i,j) - B(i,j)\right)^2 \tag{3.5.3}$$

for an $N \times N$ block.

Assuming that the basis and fractal functions are orthonormal, the error in the collage approximation is

$$\left\| G - \sum_{i=1}^{n} c_i B_i - eH \right\|^2 = \|G\|^2 - 2\sum_{i=1}^{n} c_i \langle G, B_i \rangle + \sum_{i=1}^{n} c_i^2 - 2e\langle G, H \rangle + e^2 , \tag{3.5.4}$$

where G is a discrete block from the image we wish to approximate, H is a contracted parent block and $\{B_i\}$ the discretized version of the basis.

Proof:

$$\begin{aligned}
&\left\| G - \sum_{i=1}^{n} c_i B_i - eH \right\|^2 \\
&= \|G\|^2 - \sum_{i=1}^{n} \langle G, c_i B_i \rangle - \langle G, eH \rangle \\
&- \sum_{i=1}^{n} \langle c_i B_i, G \rangle + \sum_{i=1}^{n}\sum_{j=1}^{n} \langle c_i B_i, c_j B_j \rangle + \sum_{i=1}^{n} \langle c_i B_i, eH \rangle \\
&- \langle eH, G \rangle + \sum_{i=1}^{n} \langle eH, c_i B_i \rangle + \|eH\|^2 \\
&= \|G\|^2 - 2\sum_{i=1}^{n} c_i \langle G, B_i \rangle + \sum_{i=1}^{n} c_i^2 - 2e\langle G, H \rangle + e^2
\end{aligned}$$

So, of course, if all coefficients are zero the error is $\|G\|^2$. The decrease in this error due to basis function i is

$$2c_i\langle G,B_i\rangle - c_i^2 \tag{3.5.5}$$

and for the fractal term

$$2e\langle G,H\rangle - e^2 . \tag{3.5.6}$$

As a result the effects of quantization can be easily measured.

In general a coefficient c with quantization factor Q, function B, bin number q and reconstructed value c' gives a decrease in collage error using the quantized coefficient of

$$2c'\langle G,B\rangle - c'^2 . \tag{3.5.7}$$

Let $c' = c + \delta$, then

$$\begin{aligned} &2c'\langle G,B\rangle - c'^2 = 2(c+\delta)\langle G,B\rangle - (c+\delta)^2 \\ &= 2c\langle G,B\rangle - c^2 - \left(-2\delta\langle G,B\rangle + 2c\delta\langle G,B\rangle + \delta^2\right) \end{aligned}$$

Since $c = \langle G,B\rangle$ (see equation 3.4.8)

$$2c'\langle G,B\rangle - c'^2 = 2c\langle G,B\rangle - c^2 - \delta^2 . \tag{3.5.8}$$

Hence the loss in approximation quality due to quantization is simply δ^2.

Assuming the probability distribution $p(x)$ of the coefficient is locally linear, the mean value of δ is $Q \cdot N/4$ and the mean loss due to quantization

$$\frac{N^2}{12}Q^2 . \tag{3.5.9}$$

Proof: Assuming the bin is centred on zero,

$$\int_{x=-Q\cdot N/2}^{Q\cdot N/2} p(x)\cdot x^2 dx = \frac{2}{Q\cdot N}\int_{x=0}^{Q\cdot N/2} x^2 dx,$$

since $p(x)+p(-x)=p(0)$. Hence

$$\frac{2}{Q\cdot N}\int_{x=0}^{Q\cdot N/2} x^2 dx = \frac{2}{Q\cdot N}\left[\frac{x^3}{3}\right]_0^{Q\cdot N/2} = \frac{N^2}{12}Q^2.$$

Chapter 4

METHOD

The implementation details of a fractal image compression scheme have a significant effect on the performance of the scheme. A set of parameters for the BFT are selected based on previous work (section 4.1). The details of quantization, entropy coding, and partitioning are explained in sections 4.2-4.4.

In the studies presented in later chapters decisions must be made between new parameter choices. The basis for these decisions will be their comparative performance in practice and consequently an effective method for comparison must be developed and justified (section 4.5). In the development of new techniques it will also be necessary to determine which technique performs best in practice. This comparison method is used here also.

4.1 BFT Parameters

The conclusions of Woolley [81] are used as a foundation for the extensions of and improvements to the BFT in this work. The following describe the main parameter choices which have been shown in previous work to give the best implementation.

Searching

Examination of the value of searching has showed it to be less effective in overall rate-distortion than the non-searching case [82]. A similar result holds for block isometries [82], confirmed by Saupe [69]. Although more advanced methods of searching exist, e.g. [9], and some of these may provide competitive performance, this thesis develops the non-searching avenue of investigation which has been successful so far.

Centred parent

The study of searching [81,ch.5] shows that the probability density function of parent locations is simple, with probability of a match decreasing as distance from the child increases. The most probable parent location is that of a parent centred on the child.

Consequently when using a fixed parent location a slightly better approximation can be obtained by using a centred parent for each child rather than the original parent-child arrangement first introduced in [49], where an image block was divided into four children and was used as the parent for each child. A centred parent location has been used for fractal coding by Bethel [8,Ch3].

Unfortunately to render a single centred parent block a larger part of the approximation must be computed at the same time, whereas a parent block in the original arrangement can be computed independently. However this is a relatively minor disadvantage. A comparison of the rate distortion effects of using a centred parent has been performed although has not been published other than in [76]. The results show a small gain in fidelity at the same bit rate across a range of images. The highest increase observed was 0.44 dB at 0.4 bits-per-pixel on the Boats image.

Polynomial Basis

Woolley [81,Ch.3] compares the effects of using different sets of basis functions (3.2.2) for a full range of fixed block size partitions and degrees of quantization.

The sets compared are bi-linear, $p_k(x,y) = c_{k,0} + c_{k,1} 2\sqrt{3}x + c_{k,2} 2\sqrt{3}y$, bi-quadratic, $p_k(x,y) = c_{k,0} + c_{k,1} 2\sqrt{3}x + c_{k,2} 2\sqrt{3}y + c_{k,3} 6\sqrt{5}\left(x^2 - 1/12\right) + c_{k,4} 6\sqrt{5}\left(y^2 - 1/12\right)$, etc. When results for different orders, degrees of quantization, and block sizes were compared it was found that the order 2 (bi-quadratic) transform with light

quantization produced the best results for all block sizes. This effectively proves that the bi-quadratic transform is optimal for a quadtree implementation.

Consequently the bi-quadratic basis is adopted for the work in this thesis. Incidentally the effects of using an equivalent cosine basis have been examined [76,8,p75], and a cosine basis was in fact used in [77]. The coding performance of a cosine basis is essentially identical to that of the polynomial equivalent.

Quadtree Partition

The results of Woolley [81,Ch.8] show that quadtree partitions perform significantly better than fixed block size partitions. H-V partitions were also examined but were shown to be generally inferior to quadtree partitions.

Decoding

For simplicity iterative decoding is used, rather than pyramid decoding (see section 2.6.) This gives exactly the same reconstruction as other methods and so is adequate for the study of the techniques presented in this thesis. A pyramid decoding scheme could be implemented if greater speed was required.

4.2 Quantization

The method of uniform quantization described in section 3.5 will be applied to transform coefficients.

The effect of quantization is the same for each coefficient, so if each basis function and the fractal function are all equally important $Q_i = Q_f = Q$ is constant for all i. Different coefficients will have different probability distributions, with the higher order basis functions being less useful and having a lower variance. In principle the difference in distributions will be taken into account at the entropy coding stage. However in practice, as illustrated by other work which has studied quantization in fractal schemes, for example [53],

it has been found to be more effective to quantize the higher order basis functions more coarsely. This issue will be examined in later chapters.

The use of the block size in the quantization step (equation 3.5.1) means the coefficients have quantization error proportional to the number of pixels in the block. This means the quantization error is distributed evenly throughout the image and a good visual result is produced. Although this is not optimal in purely rate-distortion terms, if the block size is not used in the quantization step the distributions of a coefficient would vary in standard deviation between block sizes, making entropy coding more difficult. For example, a block of size mN will have a probability distribution with standard deviation m times greater than that for a size N block.

4.3 Entropy Coding

The method of entropy coding employed is Huffman coding, which is explained in section 1.5.3. For more detail see [28]. Whilst no comparison of methods of entropy coding is performed, Huffman coding is widely regarded as giving good performance for fractal coding and has been used in previous work on the BFT [81,8] and other fractal coding schemes [19].

The Huffman algorithm itself is straightforward, but optionally the Huffman tree may be generated in advance using a large sample of coefficients or some model of the coefficient distributions. Both cases have the advantage that the Huffman table is constant and does not need to be included in the compressed form of the image. However neither gives an optimal compression of the source data.

No examination of this problem has been carried out since it should not affect the comparative results of methods in this thesis. All Huffman tables will be stored as part of the compressed image.

4.4 Quadtree Partitioning

Quadtree partitioning begins in this work by tiling the image with a set of *macro blocks* of size 32x32, with smaller blocks at image edges if the image dimensions are not multiples of 32. Partitioning proceeds by either estimating or finding the approximation error in each block. The block with the highest error is then split into four squares of equal size. The error value of the smaller squares is recalculated and the new blocks are added to the partition. The process of splitting the block with the highest error continues until some criterion for its termination is met. Typically this criterion is that the maximum block error falls below a certain fidelity threshold or that some maximum number of allowed blocks is reached, but it is possible to have more advanced criteria, such as calculating or estimating the bit cost of the encoding so far and terminating the process if the target compression ratio has been reached.

Several criteria were examined by Woolley [81]:

1. Edge value: An edge image is produced by applying a Sobel mask [25] to the image. The edge value of a block is then the sum of the edge values of pixels in this block.

2. RMS error: A block is coded, quantized, and rendered. The approximation error is then used.

3. Collage error: The error in the collage approximation (see equation 3.2.4).

These criteria were compared and, as would be expected, the RMS error performed best. However it was discovered that the edge value criterion gave very similar results whilst being considerably easier to compute. The collage error performed comparatively poorly.

For much of the work presented in this thesis the RMS error cannot be used as a split criterion as its computation is too complex. This leaves a choice of edge value or collage error. Although edge value performs better it is pessimistic about the performance of fractal transforms - it assumes edge blocks are poorly approximated and smooth blocks well approximated. Whilst the latter should be true, the former contradicts most expectations of fractal block transforms. For example, Jacquin's work uses a fractal term only for edge and midrange blocks. In fact the edge value criterion is a good choice for transforms which produce only smooth approximations.

Whilst the collage error may not perform better overall it is assumed that for the purposes of comparing two block transforms, using the collage error for each one will give results which are representative of the potential coding performance of the two methods.

Storage

The quadtree is stored by constructing a binary sequence in the following manner. Each macro block is taken in turn, and its quadtree is traversed and stored recursively. A node is represented by a zero and a leaf by a one. (A node representing a split in the block, a leaf being a transformation).

To save space the binary code for the quadtree may be compressed by entropy coding or by some other method such as run-length coding. Additionally the depth of the quadtree is limited by the minimum block size possible. In this case 2x2, since the bi-quadratic terms will store this block accurately, excepting quantization error.

4.5 Comparison Method

A fractal coding method is evaluated by comparing it with other methods, and to do this its performance must be measured. Potentially the performance of the method could be measured or estimated mathematically by calculating its

performance for some general class of images, but in practice it is common to simply examine the results of applying the compression method to several *test images* at a range of compression ratios.

A fundamental drawback exists to both methods however. Whilst the compression ratio of a compressed image may be easily measured, its fidelity cannot. One formula for distortion might be

$$\frac{\sum_{i=1}^{M} d_i^2}{M}, \tag{4.5.1}$$

where M is the number of people who will view the image and d_i some rating of the distortion perceived by the i^{th} viewer. Clearly this expression cannot be calculated and so some substitute method most be found.

The best practical method for measuring distortion is to take a sample of people and ask them to rate the distortion of the image on some scale. The average distortion can then be calculated. To evaluate an image compression scheme would therefore require viewers to rate different images compressed at a range of compression ratios. This is both difficult to organise and after 20 minutes of looking at versions of Gold Hill viewers responses will almost certainly have changed! However this *psycho-visual* method is one of several useful options.

The alternative is to create a numerical measure of error which can be determined by computer. The least-square error is one widely used example (see equation 3.1.5). It is particularly easy to calculate and this gives the added advantage of being useful in mathematically optimising aspects of coding schemes. Unfortunately the square-error, which forms the basis for mean-square-error, PSNR, and SNR (equations 1.1-3), gives results which do not always correspond to the perceived error.

Other formulae for measuring error have consequently received attention in the literature [85,27]. These new formulae seek to match the perceptions of the human visual system as closely as possible and have had some success. For example in [85] PSNR is reported as having a correlation of 0.653 with subjective ratings, whereas a new "segmentation-based error metric" has a rating of 0.875. In general these new methods are hindered by the lack of understanding of the human visual system and the fact that subjective distortion depends significantly on image content.

For simplicity the square-error based measures will be used, though attention will be paid to the visual quality of pictures as well. General rules observed from the study of error measurement will be taken into account, such as that the MSE resulting from different types of distortion should be given different weights, in particular the result of [84] that blocking artefact is more annoying than blurring or ringing even if the MSE is the same.

Images used

A set of test images is used to generate comparative results. The number, type, resolution, and origin of the set of test images presents a very large number of choices. One good choice would be to use the JPEG2000 test images (see section 1.3.5), but comparisons will be done using the set of four test images used in previous BFT work, namely Lena, Gold Hill, Barbara 2, and Boats. Of these Lena and Gold Hill are the most frequently used in the literature, and consequently will receive the most attention. The originals of these images are printed on the following pages.

4.6 Compressed File Format

The structure of a compressed file is simple:

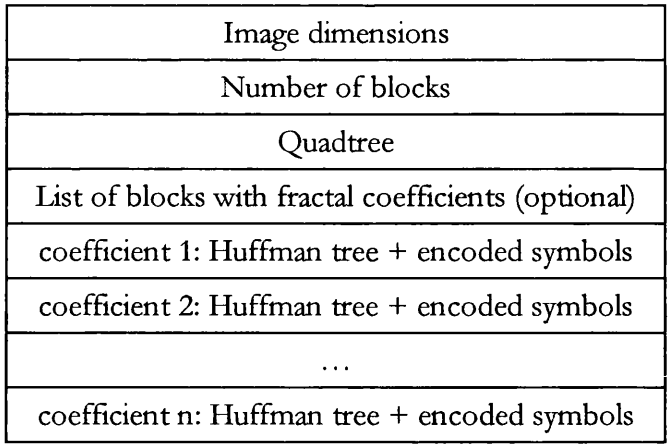


Figure 4.6.1: Structure of a compressed file

Figure 4.5.1: Lena test image.
Signal peak 226.

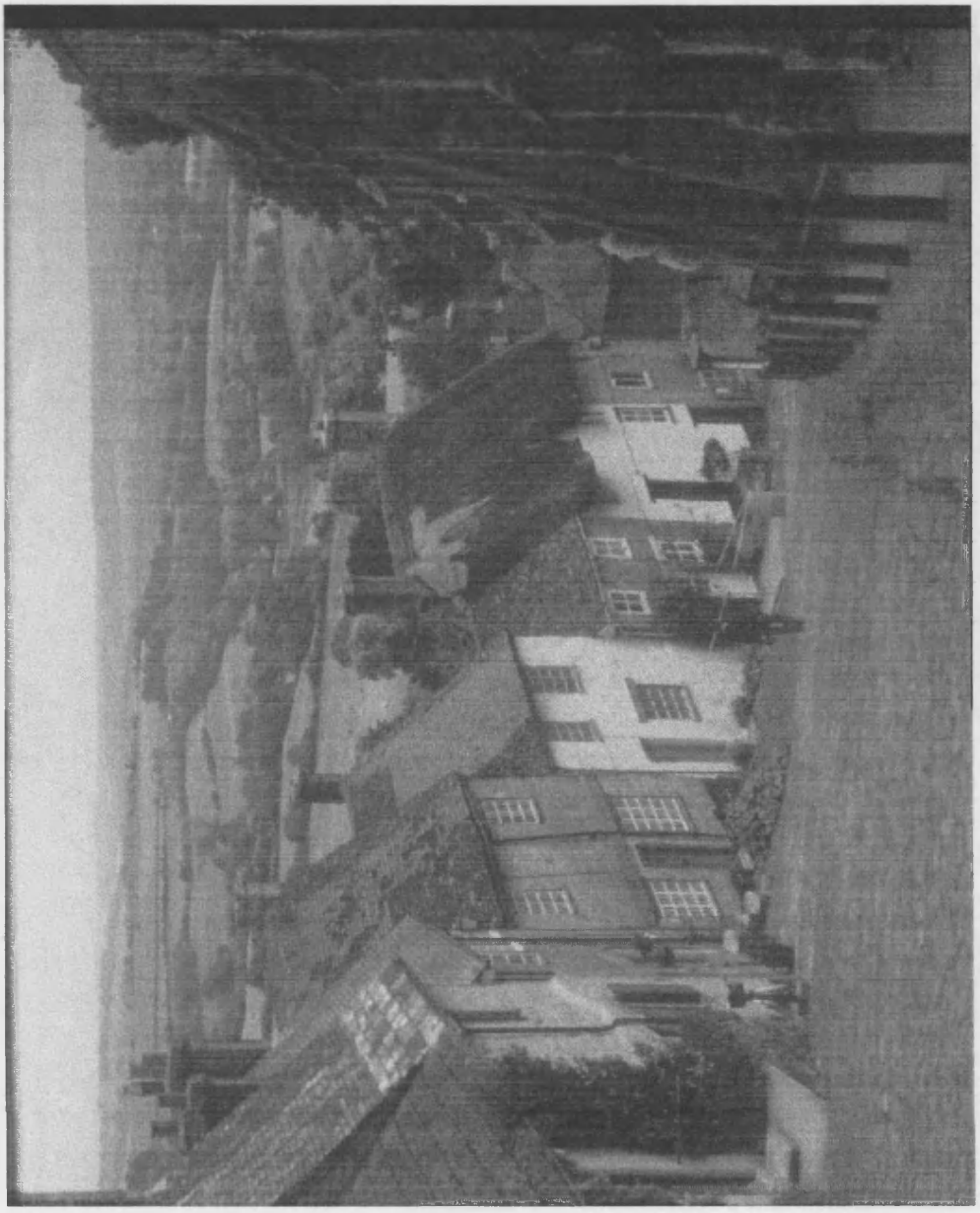


Figure 4.5.2: Gold Hill test image.
Signal peak 235.

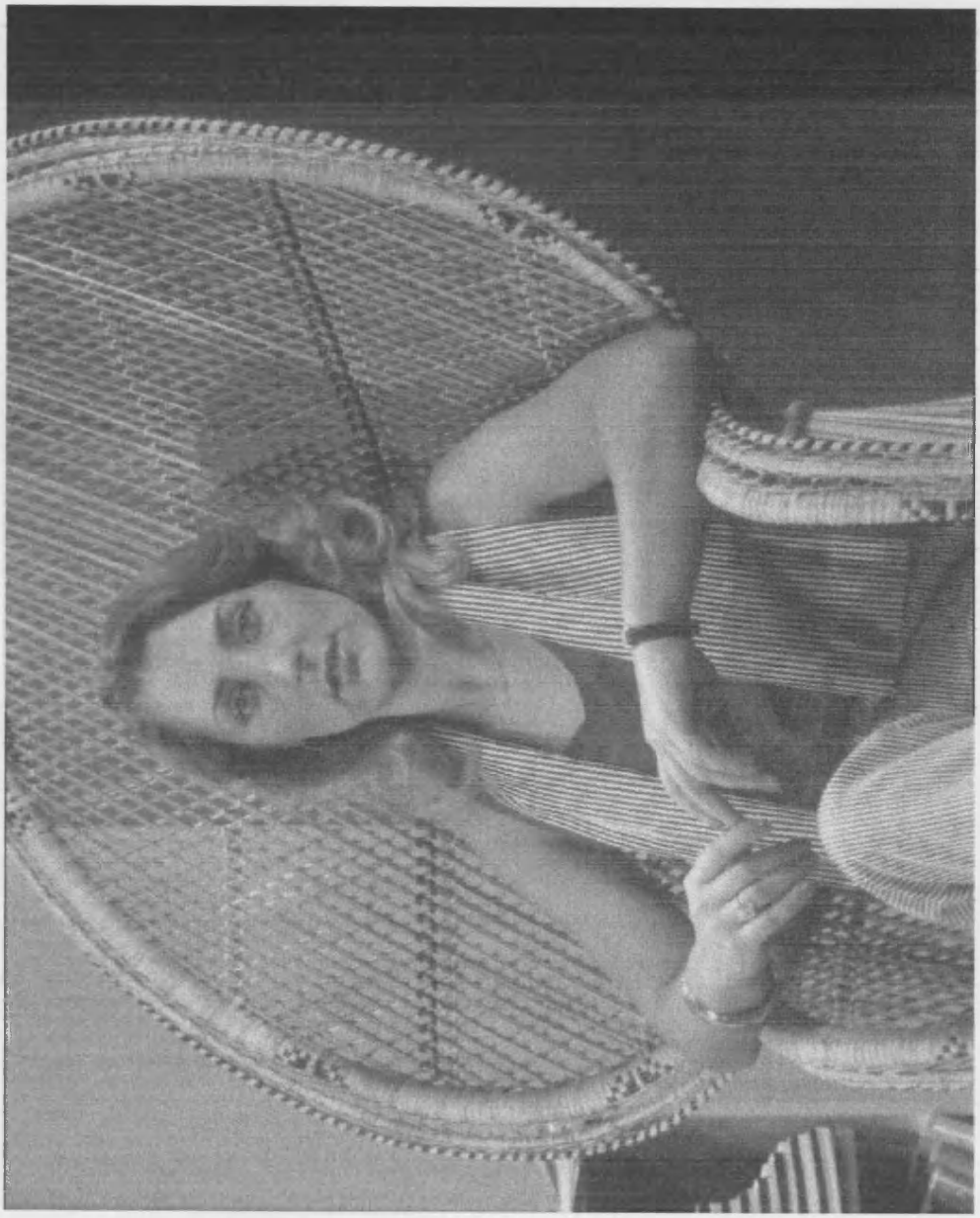


Figure 4.5.3: Barbara 2 test image.
Signal peak 235.

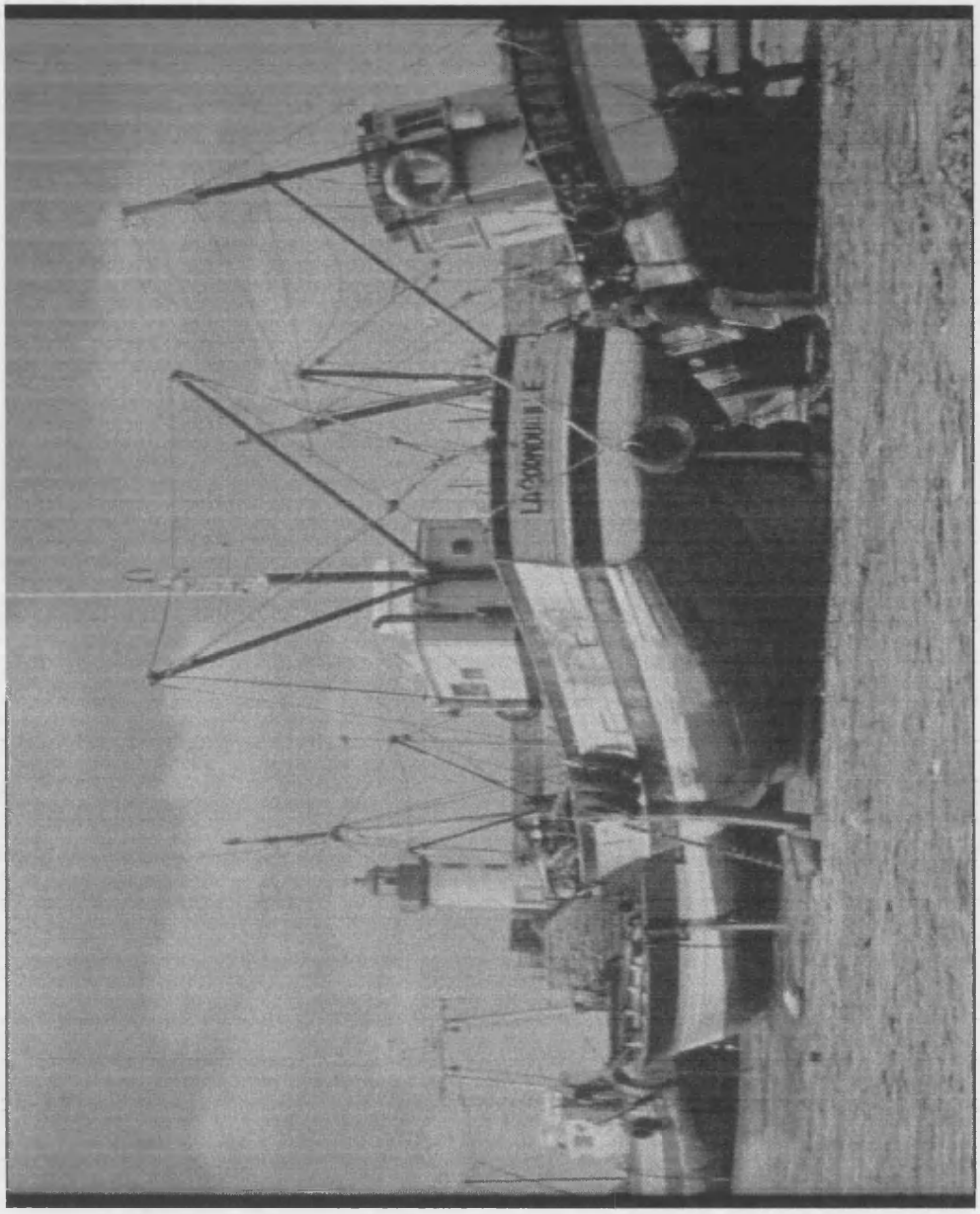


Figure 4.5.4: Boats test image.
Signal peak 220.

Chapter 5

ORTHOGONALISATION AND NORMALISATION

5.1 Introduction

Orthogonalisation and normalisation of the parent block have been proposed by Øien *et al* [60] as a means of accelerating the searching process and decoding process in fractal image compression schemes. In this chapter it is demonstrated that orthogonalisation and normalisation are of significant value for the non-searching case of the BFT, giving an improvement in rate distortion performance without extra computational cost.

As a starting point the quantization parameter $Q=4$ is used for all basis functions, which can have coefficient values as large as 256, as determined by Bethel for an orthonormal quadratic basis [8]. This parameter choice was determined as optimal by comparing alternative choices for various fixed block size partitions of the Gold Hill test image. Later work will examine the quantization of basis coefficients more closely.

The study of orthogonalisation and normalisation begins by first comparing the effects on fidelity, using fixed block size partitions and quadtree partitions with different numbers of blocks, for each of the test images, but no quantization. Secondly the distribution of the fractal coefficients is examined to see how the orthogonal and non-orthogonal coefficients differ statistically. Finally the quantization parameters for fractal coefficients in the different methods are optimised and the rate-distortion results compared.

5.2 Orthogonalisation

The formula used for orthogonalisation is given in Chapter 3. Figure 5.2.1 shows an example of orthogonalisation from the Gold Hill test image.

An orthogonal version of the BFT was implemented, using the parameters previously described. To determine the exact effects of the orthogonalisation process a number of comparisons with the original transform were made.

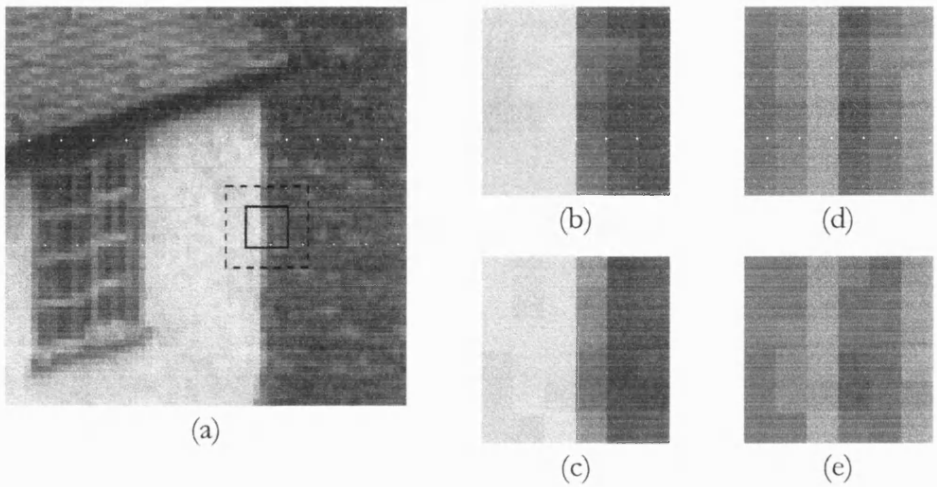


Figure 5.2.1. Orthogonalisation of image blocks. (a) Section of Gold Hill with parent and child locations marked. (b) Parent. (c) Child. (d) Orthogonalised parent. (e) Orthogonalised child.

5.2.1 Effect on Approximation Quality in Blocks

To determine the basic effect of orthogonalisation on image quality the test images were encoded with fixed block size partitions varying in size from 4x4 to 32x32. When comparing the decoded images an increase in fidelity was observed for those that had been coded by the orthogonal transform.

On average the improvement was 0.28 dB, with greater improvement for small block sizes than large ones. The effect for 4x4 blocks for example ranged from 0.44dB for the Gold Hill image to 0.95 dB for the Boats image. Figure 5.2.2 shows the PSNR results for the Boats image.

5.2.2 Distribution of Coefficients

The distributions of the fractal coefficients, derived from the 8x8 partition of Gold Hill, were examined for significant differences. As Figure 5.2.3 shows, the distributions are very similar, and the main difference is that coefficients close to zero occur more often in the orthogonal case.

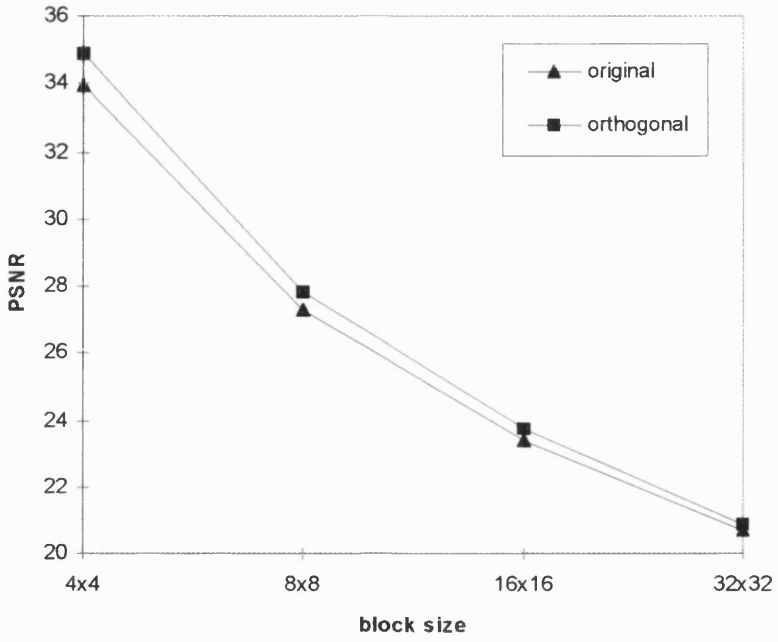


Figure 5.2.2. Results for Boats image.

Two further observations are made. Firstly that positive coefficients occur more often than negative ones, and are generally greater in magnitude. Secondly, that the non-orthogonal coefficients must be restricted in magnitude to guarantee convergence, whereas the orthogonal coefficients do not appear to require any such as restriction.

5.2.3 Quadtree Results

To compare the orthogonal and non-orthogonal transforms the test images were encoded with quadtree partitions with between 800 and 16000 blocks.

The orthogonal transform gives better PSNR figures across all test images with different quadtree partitions. Figure 5.3.1 shows a comparison of the orthogonal and non-orthogonal transforms for the Lena image.

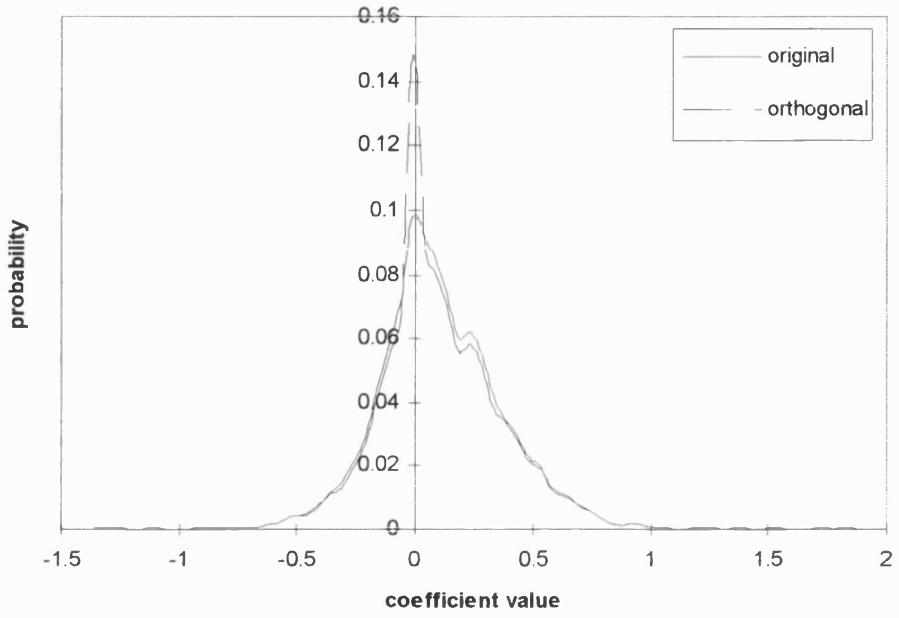
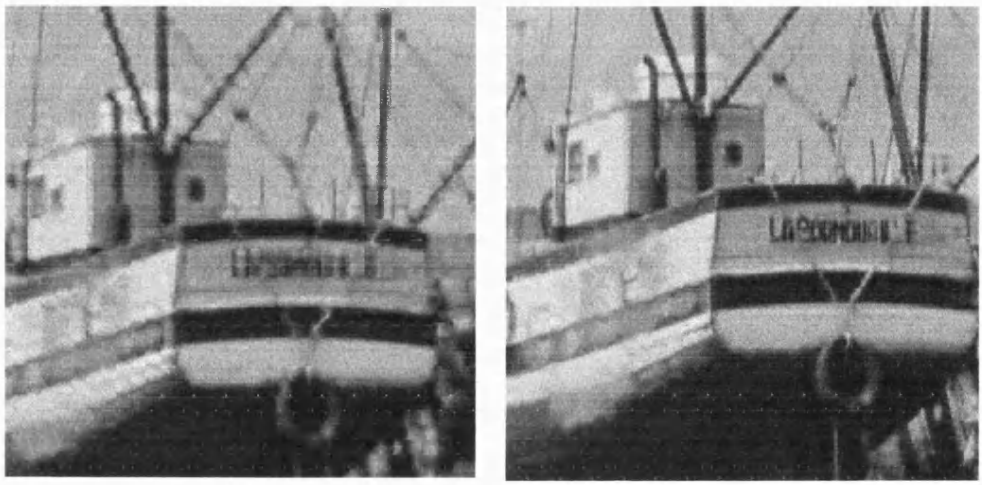


Figure 5.2.3. Probability distribution of fractal coefficients derived from Gold Hill test image with 8x8 partition.

A quadtree partition gives a very large improvement in fidelity for the same number of blocks. For comparison with a fixed block size partition see Figure 5.2.4. On average, the orthogonal transform for fixed block size partitions gave 0.28 dB improvement over the four test images, averaging 5440 blocks for Lena and 8606 for the others. Using a quadtree partition with this number of blocks, the effect of orthogonalisation was an average improvement of 0.36 dB. This increase over the fixed block size case may be assumed to be caused by the more accurate error prediction of the orthogonal transform.



(a) (b)

Figure 5.2.4 Partition results for Boats image, using orthogonal transform. (a) Fixed 8x8 block size, PSNR 27.85dB. (b) Quadtree with the same number of blocks but of varying sizes, PSNR 30.62dB.

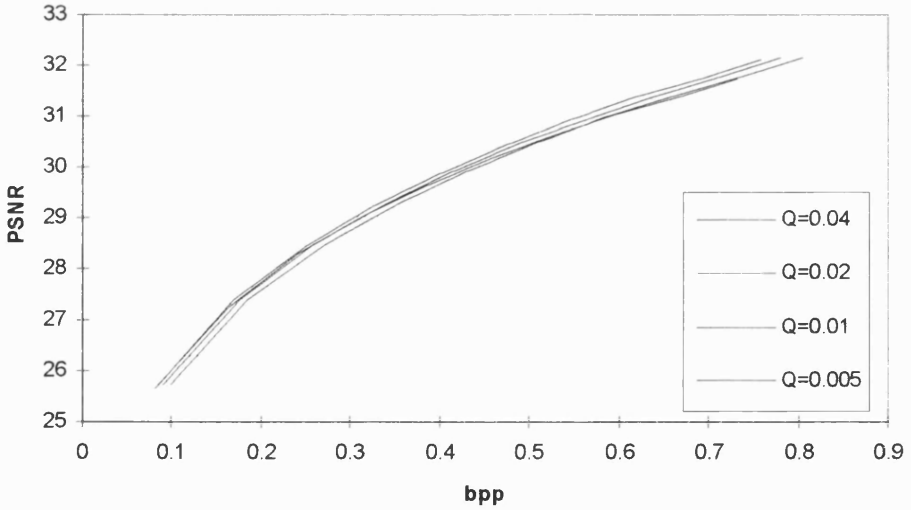


Figure 5.2.5. Quantization curves for Gold Hill test image, original transform.

5.2.4 Quantization

To determine the optimal quantization bin width for the fractal coefficients it is assumed there is a single optimal value and that the performance of the method increases as this value is approached. This assumption has been used in previous work on the BFT and other fractal methods, and experimental results indicate its validity.

Published quantization results for the original transform do not easily convert to the quantization scheme used here so the process of optimisation is repeated. This has some value however, in that a larger set of test images is used.

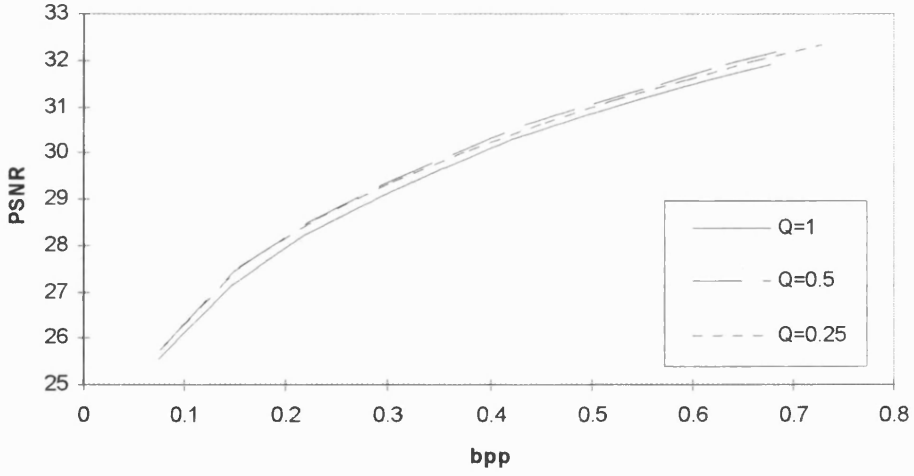


Figure 5.2.6. Quantization curves for Gold Hill test image, orthogonal transform.

The optimal quantization parameter was found to be $Q = 0.02$ for the original transform. When compared with $Q = 0.04$, $Q = 0.01$, and $Q = 0.005$, $Q = 0.02$ gave better performance for all images and bit rates. Figure 5.2.5 shows the comparison for Gold Hill.

For the orthogonal case similar results held but the best quantization was much coarser - around 0.5 for all test images, as illustrated by Figure 5.2.6, which shows Gold Hill results for 1, 0.5, and 0.25. The rate-distortion curve for 0.4 was practically identical to the 0.5 case.

The basic reason for the difference between the orthogonal and non-orthogonal transforms is that the orthogonal case has its basis components removed and so has only a residual component which is effected by the quantization.

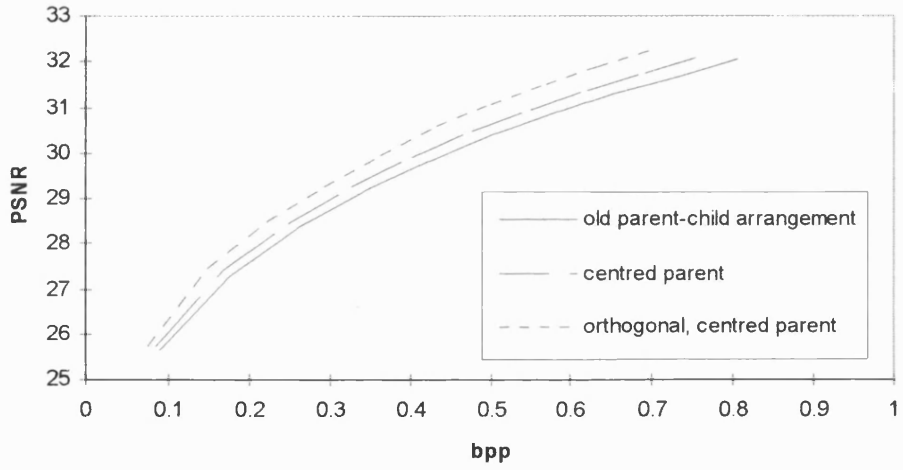


Figure 5.2.7. Comparative rate-distortion performance for Gold Hill.

5.2.5 Comparison

The use of a centred parent gives a clear improvement in fidelity over the old parent-child arrangement. Figure 5.2.7 illustrates the gain for the Gold Hill test image. However, the orthogonalisation of the parent improves the BFT still further for all test images at all bit rates. At 0.2 bits-per-pixel the average improvement over the set of test images was 0.48dB, and at 0.4 bits-per-pixel 0.53 dB. In general lower compression ratios showed a slightly larger

improvement due to orthogonalisation. Figure 5.2.8 shows example compression of the Gold Hill test image.

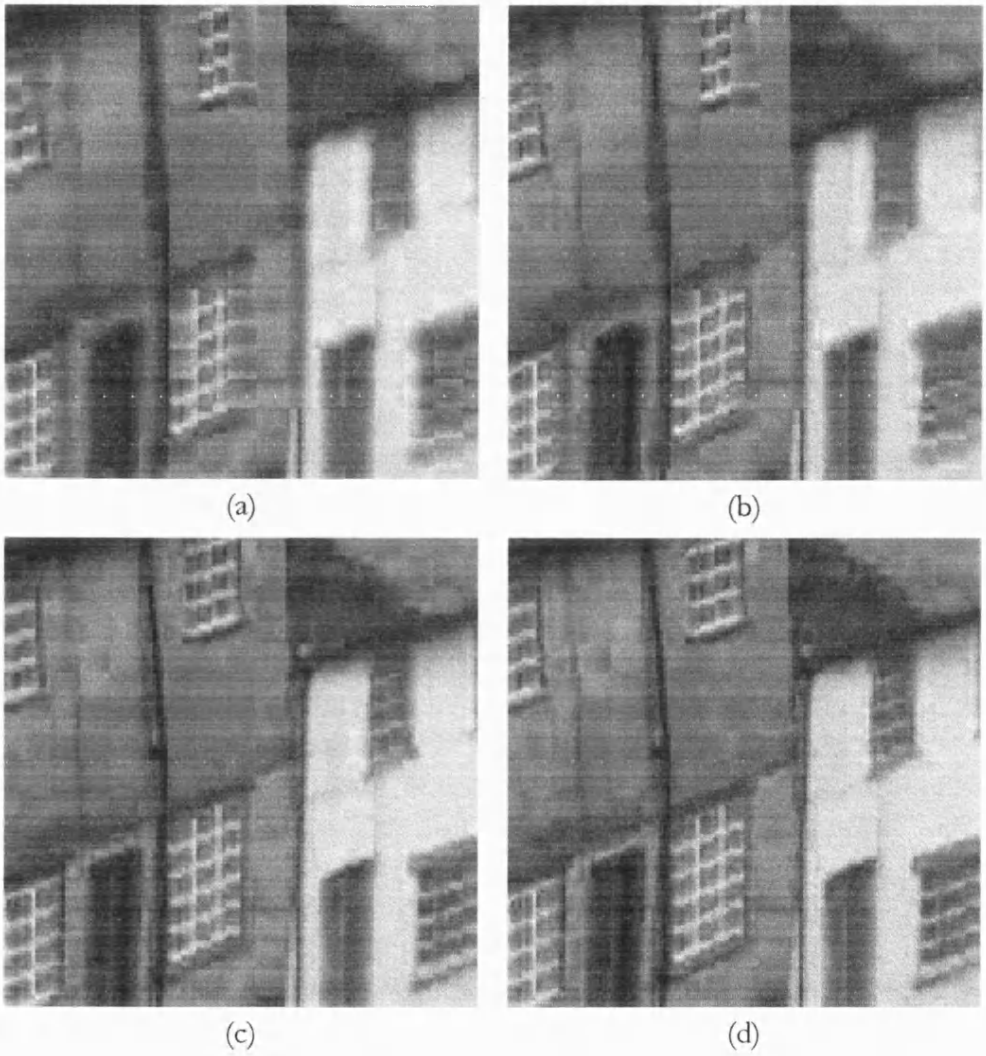


Figure 5.2.8. Compressed section of Gold Hill.
(a) 0.2 bpp original transform, PSNR 27.85dB.
(b) 0.2 bpp orthogonal transform, PSNR 28.21dB.
(c) 0.4 bpp original transform, PSNR 29.88dB.
(d) 0.4 bpp orthogonal transform, PSNR 30.30dB.

5.3 Normalisation

In the previous section blocks were orthogonal but not normalised. Consideration is now given to whether there is any further benefit to be gained by normalisation (section 3.4.2).

5.3.1 Fixed Block Size Results

The results are very similar to the orthogonal version, although there is a small improvement in every case, on average 0.04dB.

5.3.2 Quadtree Results

The quadtree results are practically identical to the orthogonal transform for all images except Lena, illustrated in Figure 5.3.1.

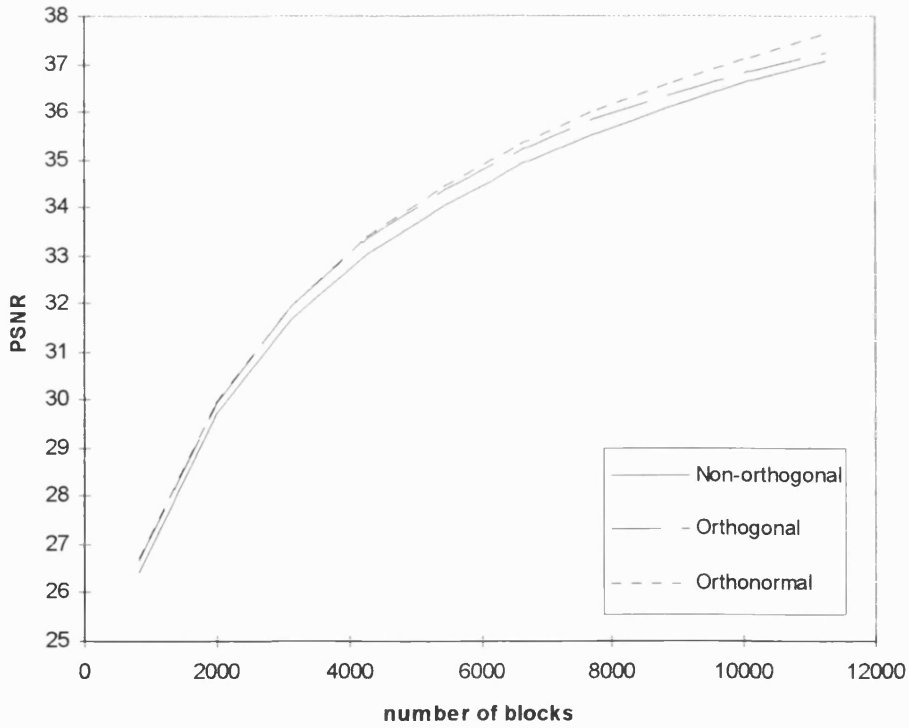


Figure 5.3.1. Quadtree results for Lena test image.

5.3.3 Quantization

Following the same procedure as in the previous section the optimal quantization parameter was determined to be $Q = 4$, which matched the results for the basis functions which are also orthonormal. This parameter choice was slightly better on all images except for Barbara 2, on which $Q = 8$ was marginally better. Overall however there was little difference between the quantization choices either numerically or visually.

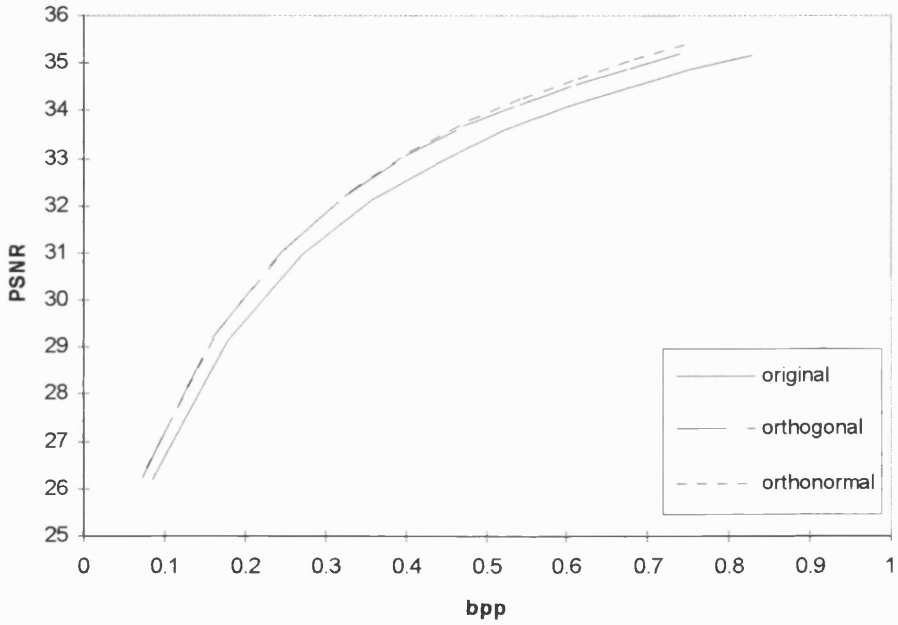


Figure 5.4.1. Comparison of transforms on the Lena test image.

5.4 Comparison of Methods

Comparing the optimal rate distortion curves of the original, orthogonal, and orthonormal methods, both the new methods are significantly better than the original transform. Between the orthogonal and orthonormal transforms there is hardly any difference. On the Lena test image the orthonormal transform

was marginally better, see Figure 5.4.1, but on other images this situation was reversed.

5.5 Conclusion

The use of orthogonal and orthonormal transforms has been studied in this chapter and the main result is that both types of transform are better than the original. There are several reasons why this is the case. From the fixed block size results the basic potential of an orthogonal transform for approximation is greater. When implemented in a quadtree scheme the comparative fidelity increases slightly further, suggesting that the orthogonal transform may be better suited to the collage based split criterion being used. From the quantization study it is clear that the orthogonal transform is significantly less susceptible to quantization error than the non-orthogonal equivalent; orthogonalised fractal coefficients may be quantized 25 times more heavily. The conclusion is that orthogonal transforms are significantly better for fractal image compression, and it may be hypothesised that this is due primarily to the removal of the DC level from the parent block, which normally dominates the working of the fractal transform.

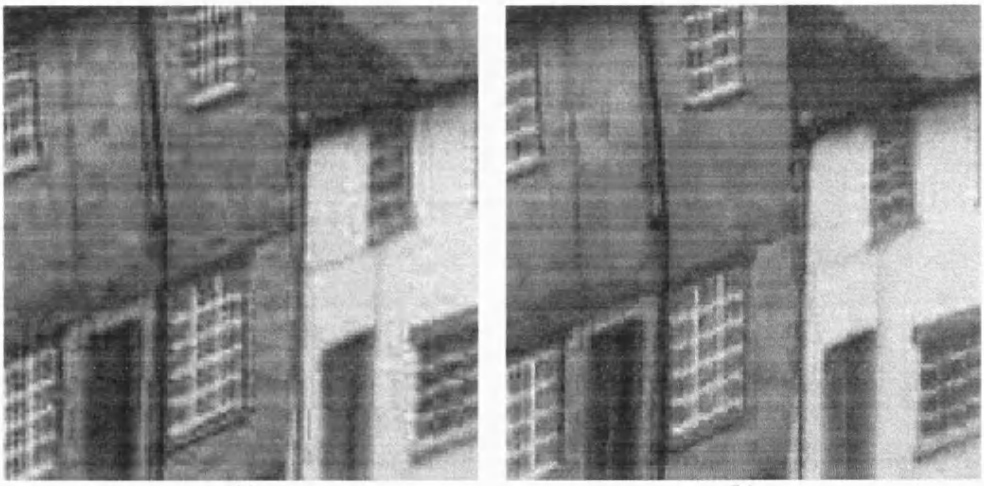


Figure 5.5.1. Gold Hill compression at 0.4 bpp.
(a) JPEG standard. (b) orthogonal transform.

Additionally, the use of a normalisation factor in the parent mapping has been investigated. It may be expected that this mapping has a disadvantage in increasing the magnitudes in parents with small norms and decreasing magnitudes in parents with large norms. The uncertainty present in the collage theorem suggests that parents with larger norms are more 'trustworthy' for the purposes of recursive mapping. However this does not appear to affect the results which show a practically identical level of performance to the orthogonal transform. A transform based on DC level removal alone may be worth investigating as a low-complexity variant of orthogonalisation, in further work. The use of normalisation adds to the decoding time of the image code, requiring approximately one extra multiplication-per-pixel per iteration.

Chapter 6

OPTIMAL QUANTIZATION

6.1 Introduction

The quantization of transform coefficients has a very great effect on the performance of the whole fractal compression algorithm. By choosing quantization parameters poorly a good technique may give rate-distortion performance far below what is possible. For this reason the problem of quantization is given special consideration here.

Although quantization has been investigated at some length in [81] and [53], the most relevant work is that of Bethel [8] which uses the same quantization scheme as this work. In Bethel's work the coefficient of each basis function is quantized with the same parameter and a comparison of $Q = 1,2,4,8$ is presented for the Gold Hill test image using a polynomial basis of order six with no fractal term. The rate-distortion curve shows clearly that $Q = 4$ is the best choice at a wide range of bit rates.

The fractal compression scheme differs from that examined in the comparison of [8] by using a fractal term, a quadtree rather than fixed block size partition, and a bi-quadratic basis. Additionally the comparison of results was only presented for the Gold Hill test image. In this chapter quantization is examined for all four test images.

The orthonormal transform previously introduced is used and, as suggested in the previous chapter, the fractal coefficient is quantized with the same quantization parameter as the basis functions.

6.2 Rate Distortion Comparison

To determine optimal quantization parameters image quality was compared at a range of bit-rates using different parameter values. Using the 0.1-1 bits-per-pixel range, rate-distortion curves were computed for each of the test images. The results[1] are shown in Figure 6.1-4.

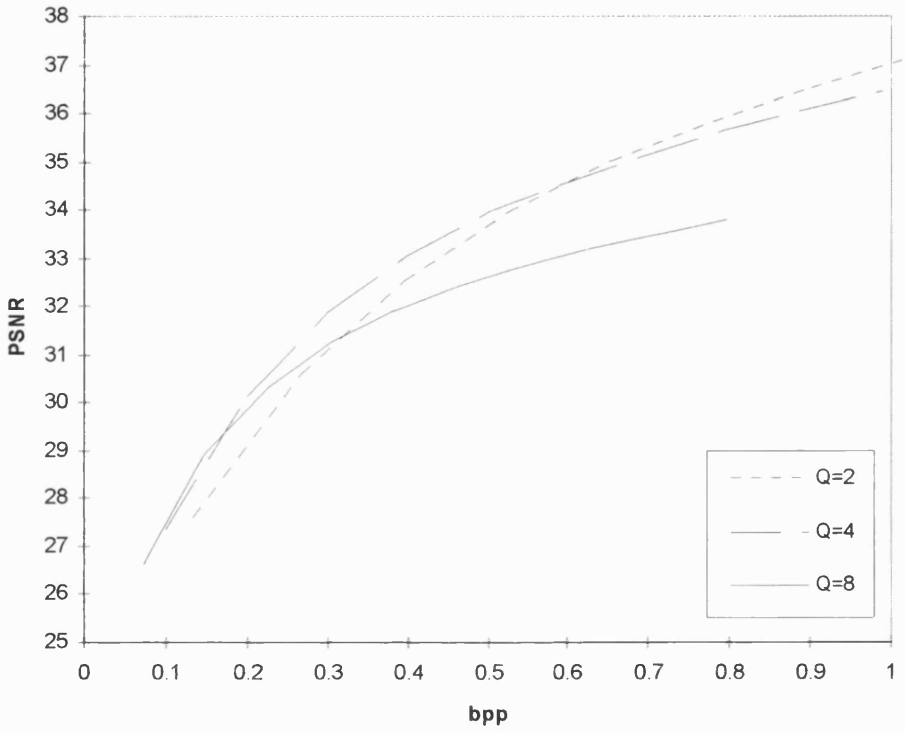


Figure 6.1. Comparison of quantization parameters for Lena test image.

[1] Barbara 2 results include the case Q=16 because high degrees of quantization gave better performance when this image was compressed.

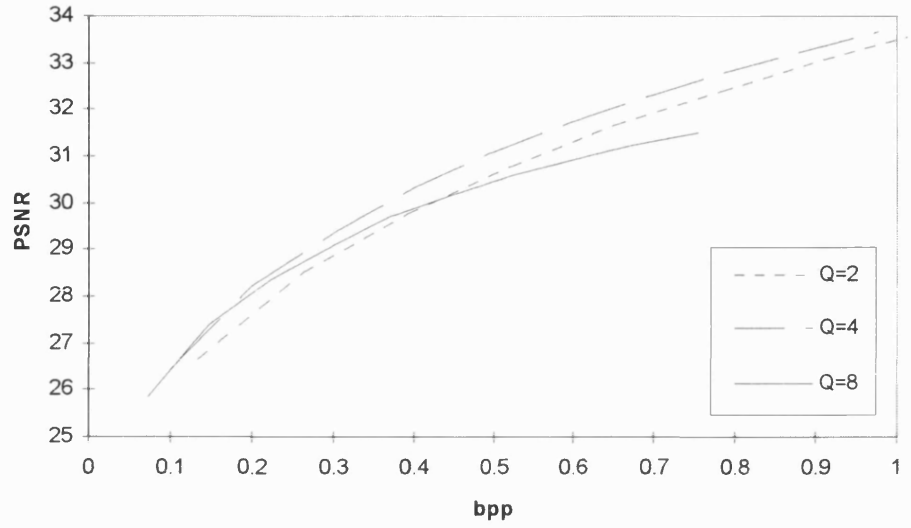


Figure 6.2. Comparison of quantization parameters for Gold Hill test image.

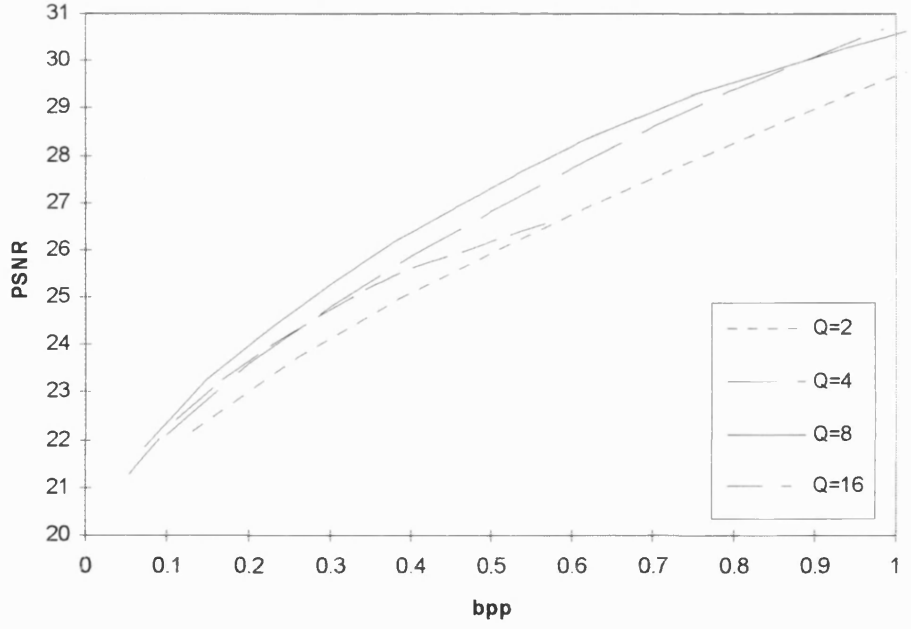


Figure 6.3. Comparison of quantization parameters for Barbara 2 test image.

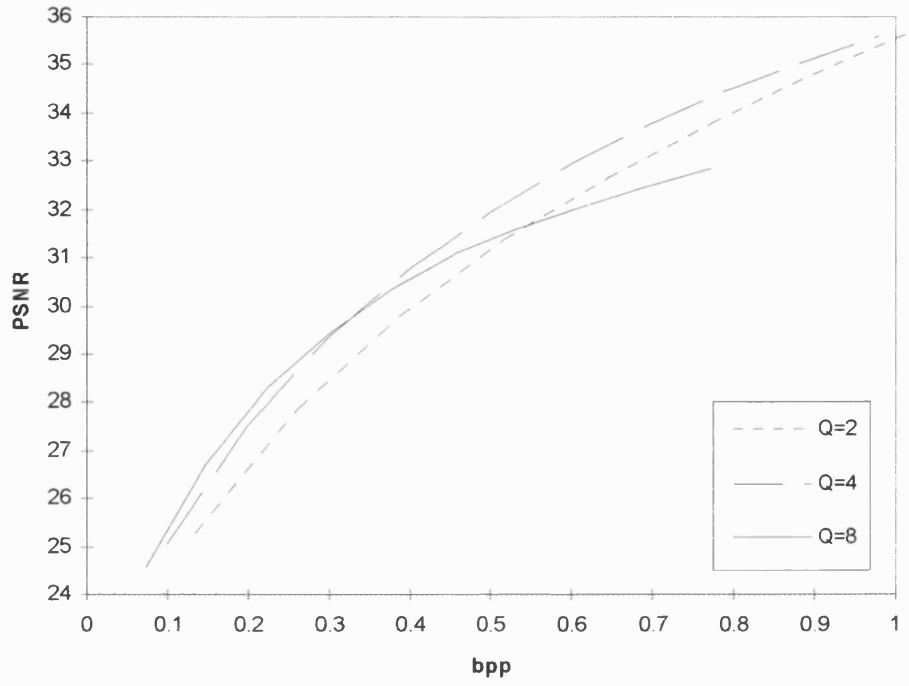


Figure 6.4. Comparison of quantization parameters for Boats test image.

No parameter value gives best performance in every case, though $Q=4$ is a good choice for the Lena, Gold Hill, and Boats images. $Q=2$ performs better at the higher bit rates for the Lena image, whereas $Q=8$ performs better at lower bit rates for the Boats image. The results for Barbara 2 are unusual in that $Q=8$ is optimal for practically the whole range of bit rates.

To conclude, optimal quantization is image dependent, but $Q=4$ is the best single choice as this generally gives very good results. However, it does not always produce the best results that the fractal compression method is capable of.

6.3 Optimal Quantization

The results of the last section show how the optimal quantization parameter value varies between images and across the range of bit-rates. This means that choosing one parameter value to use on all images will not get optimal performance from the compression scheme.

A novel solution to this problem is presented in this section. Notice that in any image the optimal quantization parameter decreases with bit-rate and that it is generally higher for images which are more difficult to approximate. E.g. Barbara 2. This suggests a dependency of the optimal quantization parameter on the bits-per-pixel (a) and image complexity (b).

Formula

Using the simplest possible formula, the quantization parameter Q can be defined as

$$Q(b,c) = a_0 + a_1 b + a_2 c\,. \tag{6.3.1}$$

To measure the complexity of an image the following function of blocks of four pixels is introduced:

$$EV(p_1,p_2,p_3,p_4) = |\bar{p} - p_1| + |\bar{p} - p_2| + |\bar{p} - p_3| + |\bar{p} - p_4| \tag{6.3.2}$$

where $\bar{p}$ is the average of the pixel intensities p_1, p_2, p_3, p_4. The complexity of an image is then quantified as the average of the above measurement of edge value applied to all blocks of four pixels in the image.

To determine the a_0, a_1, a_2 values in (6.3.1) a subset of the results in the previous section is taken and used in a least-squares optimisation. I.e. a set of data points $\{(Q_i, b_i, c_i); i = 1,\ldots,n\}$ is chosen, where Q_i is the best quantization parameter value observed at bit rate b_i, for a test image with complexity c_i.

This subset of results was chosen intuitively from amongst the test images and rates of compression used.

The result of the optimisation was:

$$3.1355+0.2404b-4.4513c\,. \qquad (6.3.3)$$

Using this formula the quantization parameter is determined at the time of encoding from the target bit-rate and the complexity of the image.

Results

To evaluate the method rate-distortion curves were again produced for each of the four test images. The results are shown in Figures 6.5-8.

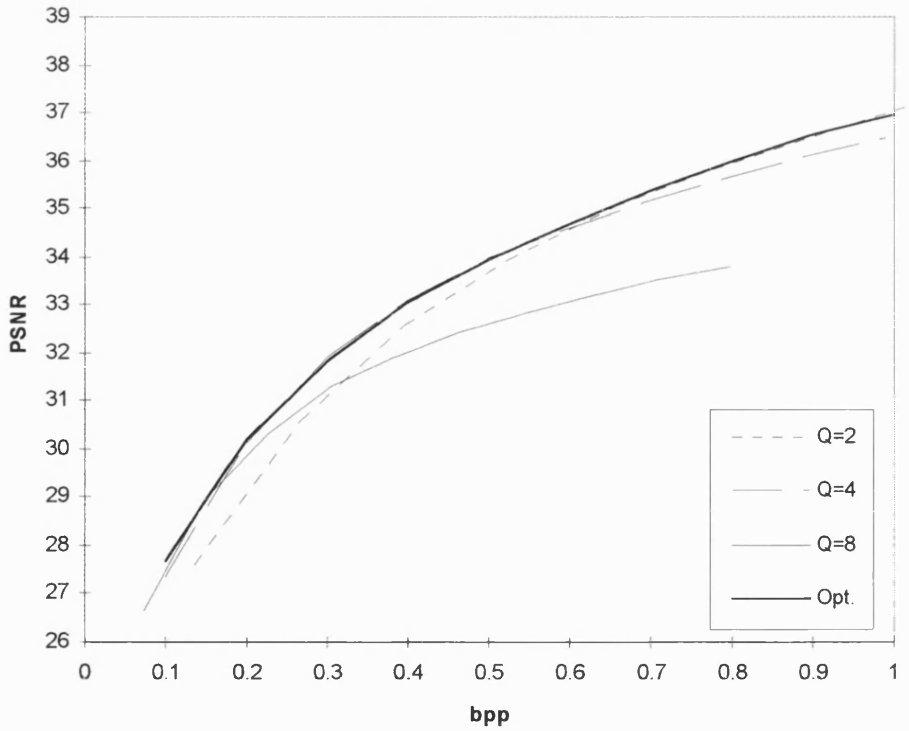


Figure 6.5. Optimal quantization for Lena test image.

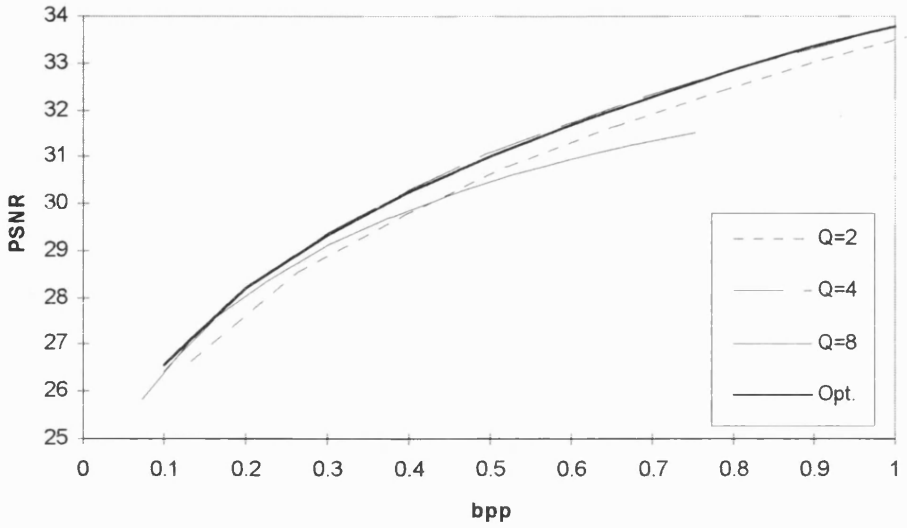


Figure 6.6. Optimal quantization for Gold Hill test image.

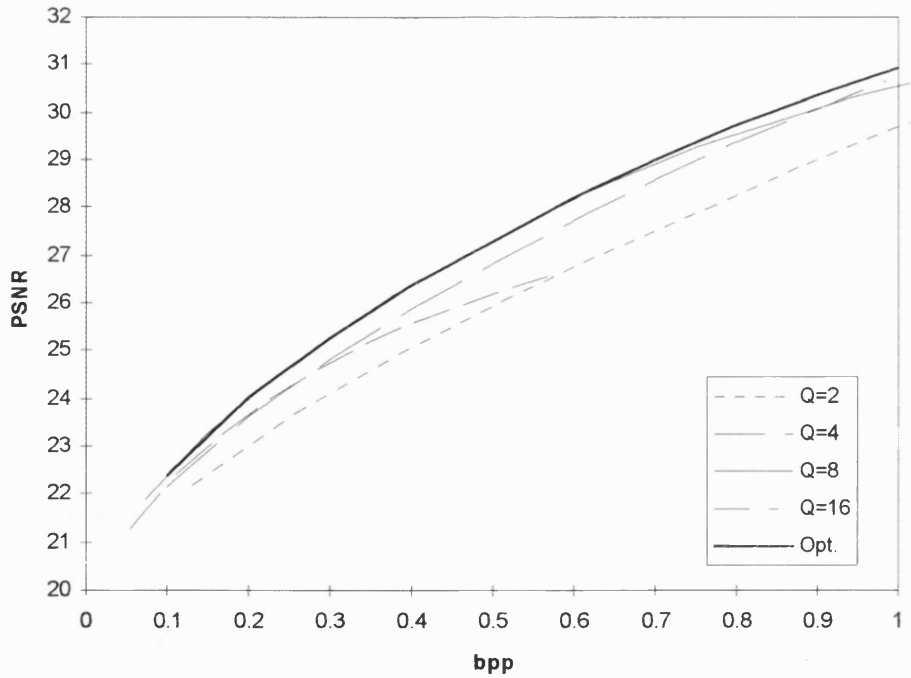


Figure 6.7. Optimal quantization for Barbara 2 test image.

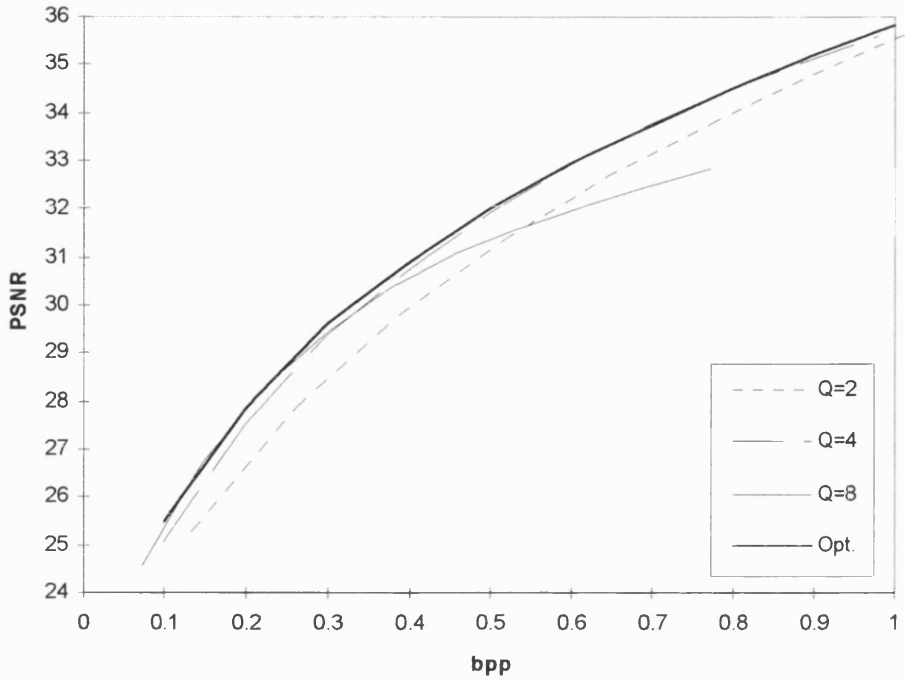


Figure 6.8. Optimal quantization for Boats test image.

The formula for the quantization parameter gives optimal performance or near-optimal performance on every image. Compared with the previously used choice of $Q=4$ this is a significant improvement. The Lena image compressed at 0.9 bits-per-pixel, for example, has a PSNR 0.46 dB higher with the new quantization scheme. On the Barbara 2 test image, for which $Q=2$ was the best choice the formula gives results which are as good as this for bit rates below 0.6 bits-per-pixel, and better at higher rates. Similarly for the Boats image where different parameter choices are optimal at different bit rates the formula gives results which match the optimal choice at any rate.

6.4 Psycho-visual Trial

As explained in section 4.5, PSNR is an imperfect measure of image quality. An important method for judging the merits of different transform options is the

psycho-visual trial, where a sample group of people identify preferred images from those produced under different options.

Quantization level	Gold Hill		Barbara 2	
	PSNR	Rating	PSNR	Rating
Q=2	26.43	10	33.78	28
Q=4	27.86	11	34.44	7
Q=8	28.33	15	32.59	0

Figure 6.9. Psycho-visual results, showing PSNR and the number of people who thought each image was best.

In the case of quantization it is useful to have an idea of the correlation between PSNR results and observed quality. To achieve this two sets of images were produced showing image sections of Gold Hill and Barbara 2 at the same bit rate with different quantization options. A group of volunteers were asked to select the best image from each group. The results are shown in Figure 6.9, and the images used in Figures 6.10 and 6.11.

(a) (b) (c)

Figure 6.10. Gold Hill image section compressed at 1 bpp. (a) Q=2. (b) Q=4. (c) Q=8.

The results show that there is a basic correlation between PSNR and observed quality for the Gold Hill image. The rating of an image was higher with the PSNR, although all images were relatively popular and there was no clearly preferred option. For the Barbara 2 image the extra blocks given by higher quantization did not impress viewers and the lowest quantized image proved

the most popular. This was not in agreement with the PSNR, but the PSNR for this image was relatively high.

6.5 Summary

This chapter has presented an investigation of the quantization parameters for different test images. Graphs of the rate-distortion performance of the fractal compression scheme under different parameter values show the importance of this area to the attainment of good overall performance.

Section 6.2 compared a number of possible parameter values over the four test images in a similar way to that of [8], and it was seen that the optimal parameter choice depended both on the degree of compression and the image being compressed.

A method was then introduced which determined the parameter value automatically. Using a formula derived from the previous results, optimal performance of the method was obtained consistently over all test images and bit-rates.

(a) (b) (c)

Figure 6.11. Barbara 2 image section compressed at 1 bpp. (a) Q=2. (b) Q=4. (c) Q=8.

A psycho-visual trial confirmed the basic correlation between PSNR and observed quality, although it should be borne in mind that this correlation was not perfect.

Overall this chapter's results are a significant step towards making fractal image compression schemes more competitive. In the case of the optimal quantization scheme, further study may prove useful. More complex formulae derived from a wider range of images may give results which are better still.

Chapter 7

RATE-DISTORTION SWITCHING

7.1 Introduction

In the previous two chapters fractal transforms were used in which every block was approximated using a basis component and a fractal component (see equation 3.2.1). However the fractal component may not always be necessary as many blocks may be smooth enough to require only the basis part of the approximation. For example, Figure 7.1 shows the basis approximation to a section of Gold Hill with an 8x8 block size partition. The basis does not approximate the windows well, but in areas where the intensities are flatter the approximation quality is relatively good.

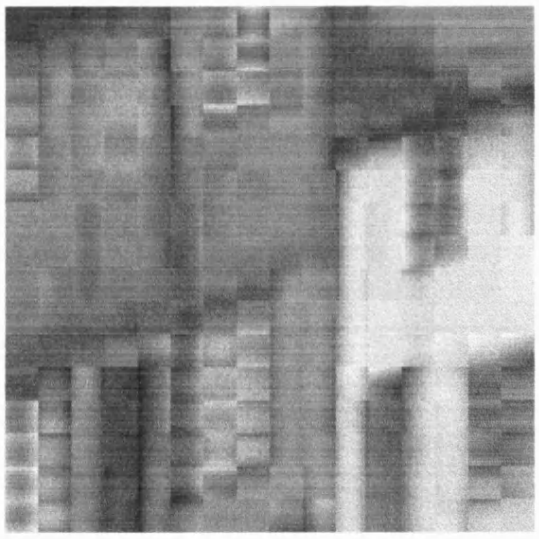


Figure 7.1. Basis approximation to a section of the Gold Hill image.

In this chapter a technique called *rate-distortion switching* is examined. This is a process whereby the fractal term is used selectively to increase compression

and improve overall rate-distortion performance. For each block in the image partition a bit can be used to tell the decoder whether that block is approximated using a fractal term or not.

Rate-distortion switching of fractal transforms was first proposed by Bethel in [77] and is also studied in [8]. It works by comparing the improvement in fidelity per bit caused by the fractal term with the gradient of the rate-distortion curve. If the improvement per bit lies below the curve the fractal term is decreasing the overall performance of the method, whereas if the improvement exceeds the curve's gradient the fractal term has a beneficial effect on the overall performance.

Selective use of fractal terms is not new and was used by Jacquin [35], though not optimally. In [8] the rate-distortion based method was used to compress the Gold Hill image, though the improvement over the normal approach was not reported. In this chapter a more thorough examination of its effects is presented, and the method is then developed through several new variations.

7.2 Theory

When encoding an image using a quadtree partition, blocks are split progressively until the encoding reaches the target bit rate. At each stage a certain number of bits have been used from the *bit budget* (the maximum number allowed.) Each time a block is split the fidelity of the approximation is improved and bits from the bit budget are used.

This process results in a rate-distortion curve which may be expressed mathematically as a continuous function $P(b)$ of the bit rate b.

$$\frac{dP(b)}{db} \tag{7.2.1}$$

is therefore the gradient of the rate distortion curve for the image being encoded. The effect of a fractal term on a block is an increase in PSNR of ΔP at a cost in bits of Δb. Therefore in essence

$$\begin{aligned}
&\frac{\Delta P}{\Delta b} < \frac{dP(b)}{db} \Rightarrow \text{fractal term decreases overall performance} \\
&\frac{\Delta P}{\Delta b} > \frac{dP(b)}{db} \Rightarrow \text{fractal term increases overall performance} \\
&\frac{\Delta P}{\Delta b} = \frac{dP(b)}{db} \Rightarrow \text{fractal term has no effect on overall performance}
\end{aligned} \tag{7.2.2}$$

7.3 Implementation

Equation 7.2.2 determines whether a block should be approximated with a fractal term or not, but several compromises are required to use the method in practice.

Firstly, the use of a fractal term in one block alters $P(b)$ resulting in a complex problem where to get the best result the use of a fractal term would have to be re-evaluated for every block whenever $P(b)$ changed. It is assumed for simplicity that the changes in the gradient are not too great. Alternatively a 'second pass' may be made when the quadtree is complete or almost complete to decide finally where fractal terms will be used.

Secondly, the value of $P(b)$ is not known for the target bit rate, it is estimated as the image is compressed. The gradient is steeper at lower bit rates so the criterion for using a fractal term is harder to satisfy earlier in the encoding process.

Thirdly, the effect of the fractal term on fidelity in a block is not known at the encoding stage, and can only be estimated using the Collage Theorem.

Fourthly, determining the bit cost when the Huffman table is only computed after the basic encoding is complete is not possible, and therefore the current theoretical entropy of coefficients is used to estimate the eventual bit cost.

To store the list of blocks with fractal coefficients a string of zeros and ones is produced, with zero for a non-fractal block and one for a fractal block. Further compression of this data may be possible, such as using Run-Length-Encoding, but this was not examined.

7.4 Results

Rate-distortion curves were generated for each of the four test images and compared. The switch increased compression and reduced fidelity but the overall effect was little change to the overall performance. For several images the switching method actually produced slightly inferior results at lower compression ratios. Figure 7.2 shows the rate-distortion curve for the Gold Hill image.

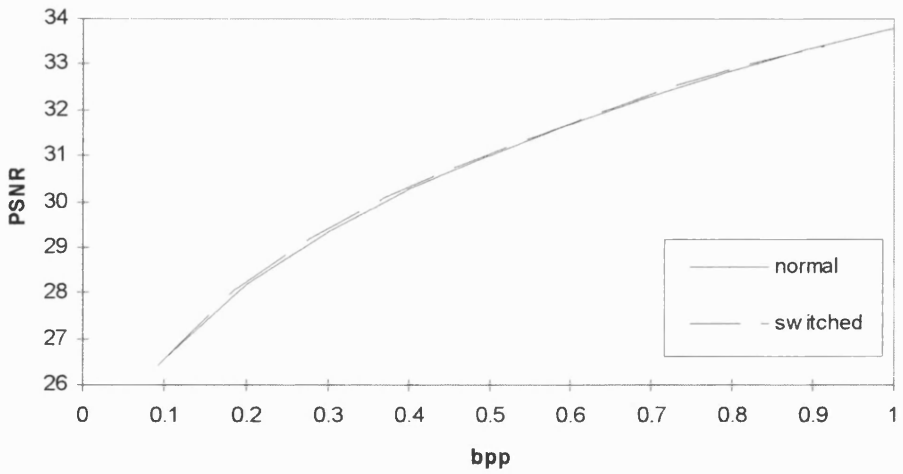


Figure 7.2. Results for Gold Hill.

Enhanced Method

The above results are not encouraging, but when switching the fractal term the final gradient of the rate-distortion curve is not known and so the method is not necessarily optimal. Once the partition is complete the gradient is known, not accounting for variations caused by the switching process itself, and the fractal terms may be switched more accurately.

This extra switching phase was implemented and the rate-distortion effects evaluated. The results show an increase in the number of fractal terms being used and an improvement in fidelity for all images.

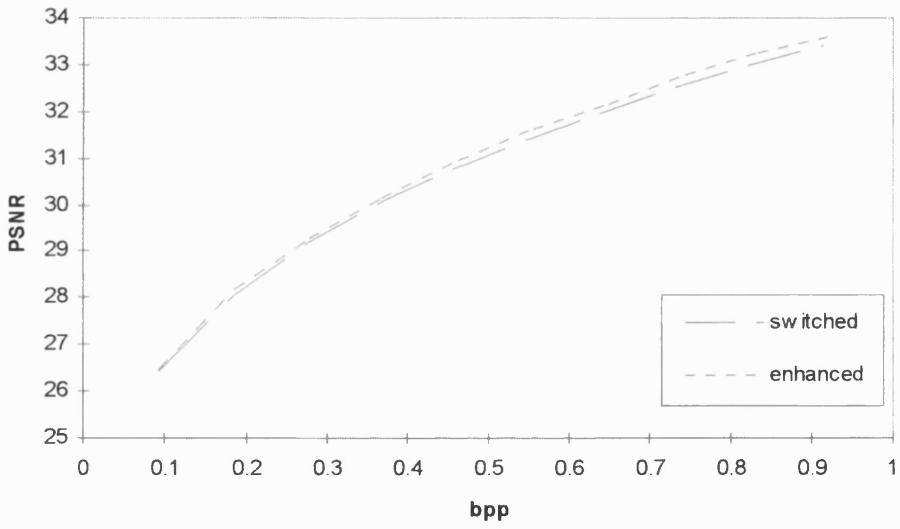


Figure 7.3. Enhanced switching for Gold Hill image.

Variations

It is not necessary to switch fractal terms on or off during the quadtree construction, and so two alternative versions of this method exist. Either a basis-only quadtree partition may be produced (Variation 1) or a partition where fractals are used in every block (Variation 2). Fractal terms can then be accurately switched on or off after the partitioning is complete.

Comparing rate-distortion results for these new methods, the first version produces inferior results, whereas the second method is similar in performance to the enhanced method and slightly better at higher bit rates on several images. It also has the advantage of being simpler to compute. Figure 7.3 shows results for the Lena image.

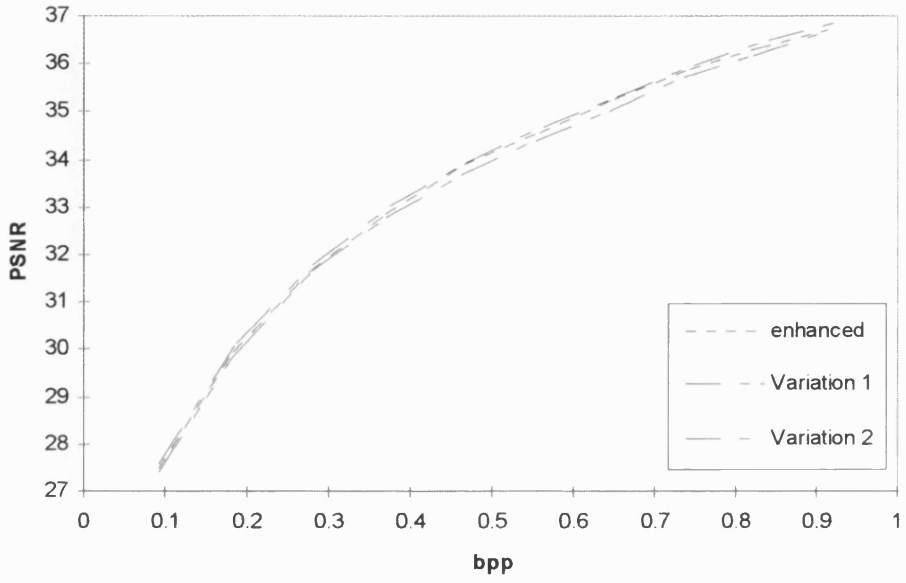
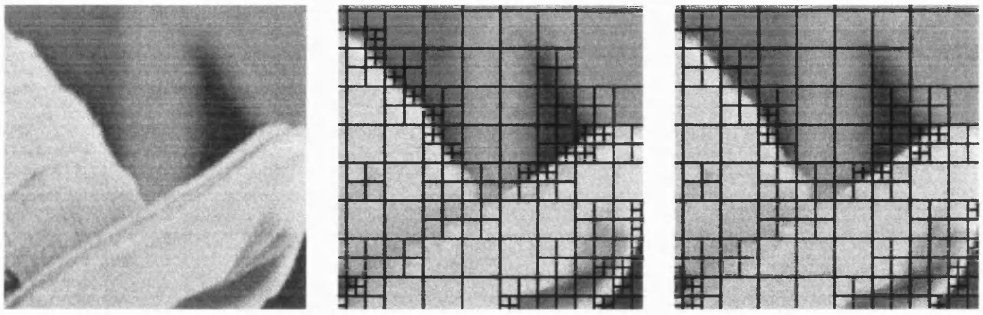


Figure 7.3. Comparison of variations for Lena image.

Note that the reason these methods differ in performance is that they produce different partitions of the image. The second variation uses a fractal term in every block and when the error is significantly lower than the basis only approximation the splitting process is effected. Figure 7.4 shows an example of this for part of the Lena image. Along the edges of the hat the second variation has not split blocks to such a small size and has used the blocks saved to improve other parts of the image. The result is that the second variation produces a partition which is closer to optimal and hence gives better overall results.



(a) (b) (c)

Figure 7.4. Quadtree partitions for section of Lena image at 0.2 bits-per-pixel. (a) Original image. (b) Variations 1. (c) Variation 2.

Usage of Fractal Terms

Adopting the second variation as the preferred method the usage of fractal terms was examined. Figures 7.5 and 7.6 show the location of blocks using fractal terms. On average fractal terms were used in the approximation of 18% of image area.

7.5 Summary

In this chapter the rate-distortion based switching technique proposed by Bethel was examined for different test images. It was found that the basic method gave no significant gain in performance - wherever bits were saved by not using a fractal term the loss in fidelity was sufficient to cancel out any advantage.

Several variations on the method were investigated and it was found that the rate-distortion switching should be carried out after the partitioning of the image. The method used in [77] corresponds to Variation 1 here, and was not the best implementation. Better results were obtained when fractal terms were initially used for every block.

Using the rate-distortion switching technique saved significantly on the bit cost. At 0.4 bits-per-pixel an average of 28% more blocks were available to the

partitioning process. The average area of the image approximated using fractal terms at this bit-rate was 18% and as expected blocks using fractal terms appeared to be those with more complex details such as edge blocks and texture blocks.

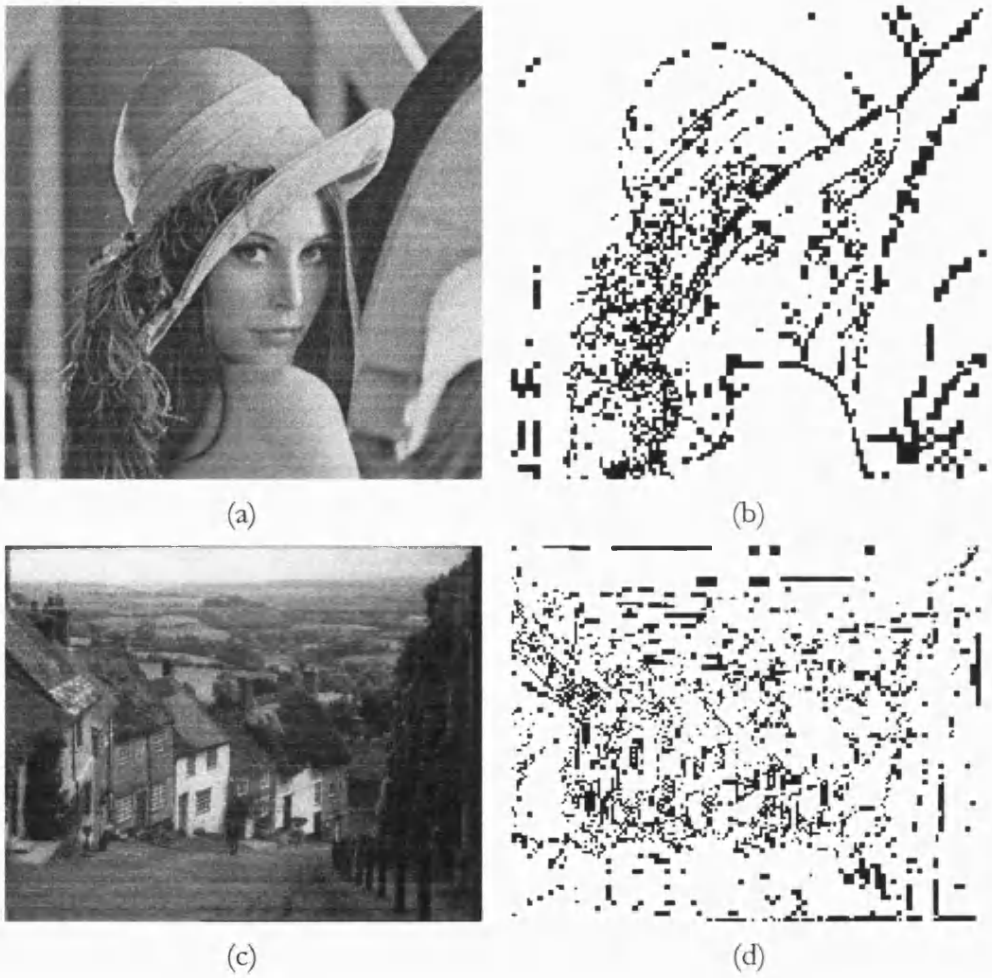


Figure 7.5. Fractal term usage for test images. Blocks with fractal terms are marked. (a) Lena original. (b) Lena fractal blocks, covering 17% of image area. (c) Gold Hill original. (d) Gold Hill fractal blocks, covering 17%.

Previously the psycho-visual aspects of the use of fractal terms have been studied by Monro and others, and these show that it may be desirable to allow more fractal terms because they are visually more pleasing, even though the PSNR decreases slightly. In their study of the Gold Hill image 21% of the

image area was covered with fractal blocks in the Variation 1 case. However when the likelihood of fractal terms being used was artificially increased, the optimal visual result was found to cover 43% of the image with fractal blocks. The maximum number of fractal terms which could be used without significant loss in fidelity was 61%.

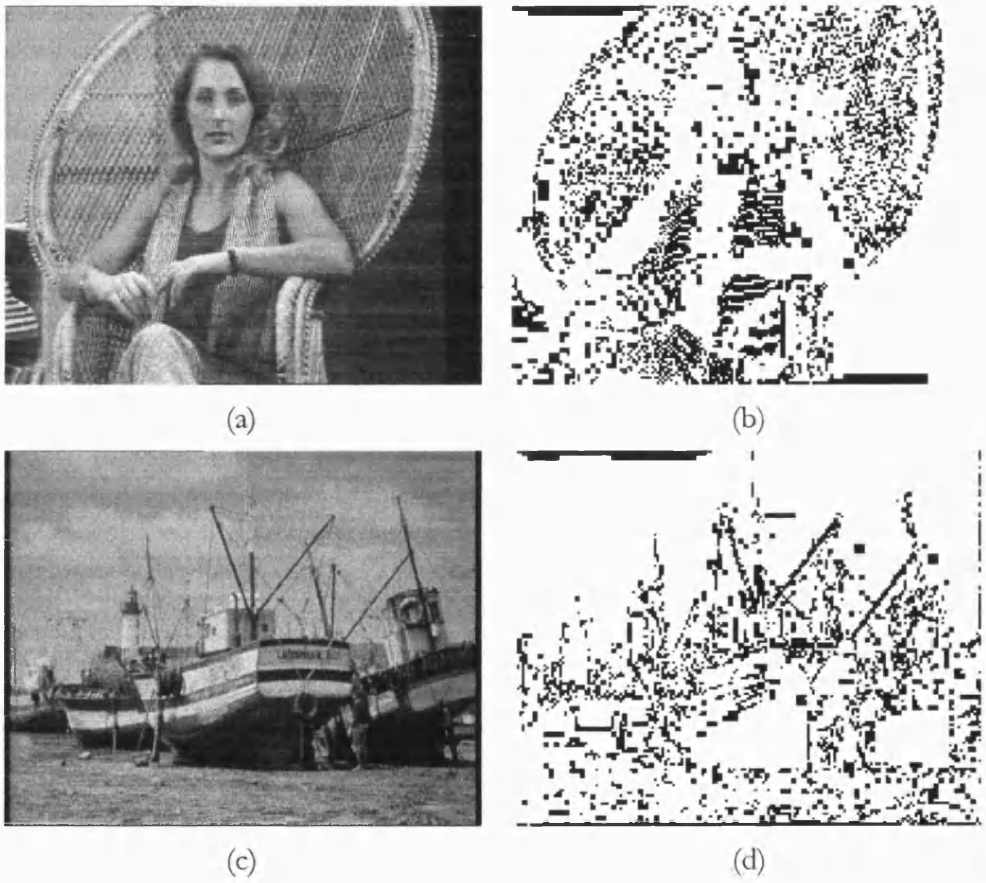


Figure 7.6. Fractal term usage for test images. Blocks with fractal terms are marked. (a) Barbara 2 original. (b) Barbara 2 fractal blocks, covering 22% of image area. (c) Boats original. (d) Boats fractal blocks, covering 17%.

Chapter 8

IMPLICIT FRACTAL TECHNIQUE

8.1 Introduction

In this chapter a novel technique is introduced which improves fidelity by computing parent locations from basis coefficients.

In much work on fractal compression the optimal parent location is found by searching, giving better image fidelity but at a high cost in bits when the location is stored. Typically the trade off between bit cost and fidelity leads to the conclusion that searching is counter productive for the overall performance of a fractal scheme [82,69].

The implicit fractal technique (IFT) presents an advance in this regard. A correlation is observed between the basis approximation and the optimal parent location. By analysing the basis coefficients and using a specially constructed edge model a method is derived in which the parent location can be determined from the basis approximation in a block without direct reference to the image - a process which can be repeated at the decoding stage and hence makes storage of the parent location unnecessary.

The IFT is a novel method which improves image fidelity in each block and has been published in [77] and [52].

8.2 Derivation

For an orthonormal bi-quadratic basis defined on $[-0.5,0.5]\times[-0.5,0.5]$

$$b_0(x,y) = 1,\ b_1(x,y) = 2\sqrt{3}x,\ b_2(x,y) = 2\sqrt{3}y, \tag{8.2.1}$$

$$b_3(x,y) = 6\sqrt{5}\left(x^2 - 1/12\right),\ b_4(x,y) = 6\sqrt{5}\left(y^2 - 1/12\right).$$

The problem is to find the parent location from the coefficients $c_0,\ldots,c_4$ with respect to the above basis.

Edge Model

A model of an edge passing through a child block is introduced, see Figure 8.1. The edge is assumed to be straight and the intensity values constant on either side. In the left-hand region, marked as region A, the intensity value is l, in the right-hand region, region B, the intensity is r. For the time being it is assumed that the edge is within 45° of the vertical, slopes to the right, and lies to the left of the centre of the child block. Other edge orientations will be dealt with later.

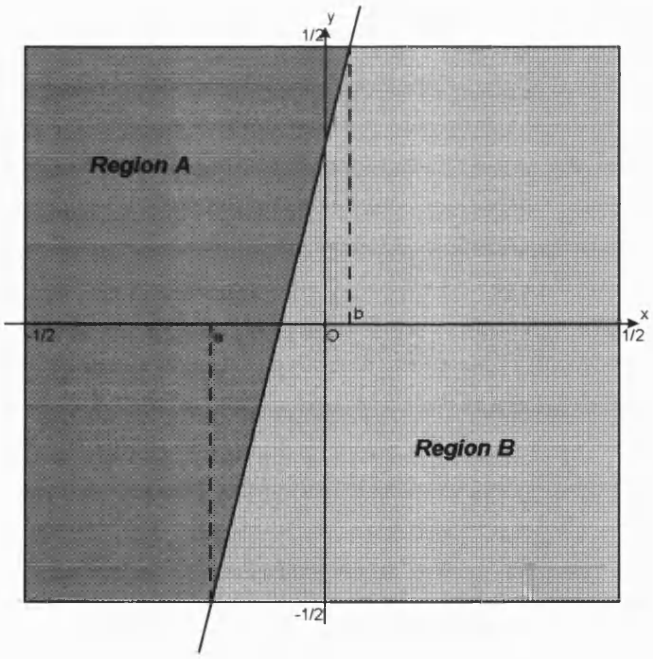


Figure 8.1. Model of a simple edge.

If the edge model matches the contents of a child block it is clear where the parent should be located. The parent should map the edge exactly onto itself as

illustrated by Figure 8.2, and so the mid-point of the edge will be the fixed point of the contraction mapping.

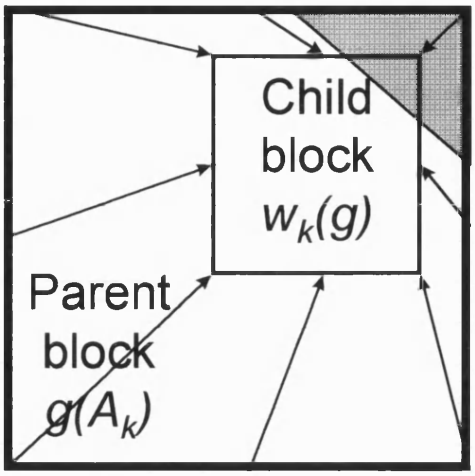


Figure 8.2. Example parent-child alignment.

The parent location can be easily determined from only the parameters a and b, by first calculating the midpoint $\underline{m}$ of the line segment making up the edge within the child block.

$$\underline{m} = \left(\frac{b+a}{2}, 0\right) \text{ if } b \geq a \geq -0.5 \tag{8.2.2}$$

$$\text{and } \underline{m} = \left(\frac{2b-1}{4}, \frac{-2a-1}{4(b-a)}\right) \text{ if } b > -0.5 > a .$$

The parent location is then computed so that the mid-point of the line is the fixed point of the contraction mapping:

$$\underline{p} = \underline{c} - k\left(\underline{m} + \left(\frac{1}{2}, \frac{1}{2}\right)\right) \tag{8.2.3}$$

where k is the child block side length in pixels and $\underline{c}$ is the child location[1].

To determine a and b formulae are derived for each of the basis coefficients c_1, c_2, c_3 in terms of the model parameters. Full details are found in Appendix B. For example, for c_1:

$$c_1 = (l-r)\left(\frac{2\sqrt{3}\left(4(b^3-a^3)-3(b-a)\right)}{24(b-a)}\right) \quad \text{for } b > a \geq -0.5. \tag{8.2.4}$$

As the parent location is independent of the intensity values l and r they are eliminated by calculating ratios of the basis coefficients. For $b > a \geq -0.5$,

$$\left|c_3 / c_1\right| = \left|\frac{\sqrt{15}\left(2(b^4-a^4)+a^2-b^2\right)}{4(b^3-a^3)-3(b-a)}\right| \tag{8.2.5}$$

$$\left|c_2 / c_1\right| = \left|\frac{2(b-a)^2}{4(b^3-a^3)-3(b-a)}\right|,$$

for $b > -0.5 > a$

$$\left|c_3 / c_1\right| = \left|\frac{\sqrt{15}(4b^2-1)^2}{8(4b^3-1-3b)}\right| \tag{8.2.6}$$

$$\left|c_2 / c_1\right| = \left|\frac{4b^3-12ab^2-12ab-3a-3b-1}{2(4b^3-1-3b)(b-a)}\right|,$$

and for $b = a$

$$\left|c_3 / c_1\right| = \left|\frac{2\sqrt{15}}{3}a\right| \tag{8.2.7}$$

[1] Block locations are specified by the co-ordinates of their bottom-left corners.

$$|c_2 / c_1| = 0 .$$

8.3 Look-up Table Solution

Using the equations 8.2.5-7 the problem may now be efficiently solved using a look-up table. The table is created using a range of possible a and b values and for each pair of values a pair of ratios, using the appropriate equation (8.2.5-7), is calculated. A new line is added to the table consisting of the ratios and the midpoint corresponding to a and b as calculated from equation 8.2.2. Figure 8.3 shows an example table based on an even distribution of a and b values.

$\lvert c_3/c_1 \rvert$	$\lvert c_2/c_1 \rvert$	a	b	m_x	m_y
0.87	1.00	-1.25	-0.25	-0.38	0.38
0.87	0.93	-1.00	-0.25	-0.38	0.33
0.48	1.00	-1.00	0.00	-0.25	0.25
0.87	0.80	-0.75	-0.25	-0.38	0.25
0.48	0.83	-0.75	0.00	-0.25	0.17
0.16	1.00	-0.75	0.25	-0.13	0.13
0.87	0.40	-0.50	-0.25	-0.38	0.00
0.48	0.50	-0.50	0.00	-0.25	0.00
0.16	0.67	-0.50	0.25	-0.13	0.00
0.00	1.00	-0.50	0.50	0.00	0.00
0.65	0.00	-0.25	-0.25	-0.25	0.00
0.31	0.18	-0.25	0.00	-0.13	0.00
0.00	0.36	-0.25	0.25	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00

Figure 8.3. Example IFT table generated for 4x4 block size.

At the encoding and decoding stages the parent location can now be determined from the basis coefficients. From the basis approximation to the block, $|c_3/c_1|$ and $|c_2/c_1|$ are calculated and the table is searched for the closest matching ratios. This gives the midpoint of the line.

```
Algorithm IFT
begin
• Compute basis coefficients, cx,cy,c2x,c2y
• { Compute ratios }
• if (cx = 0)  then begin
        • if (cy = 0) then ryx:=0; else ryx:=∞;
        • if (c2x = 0) then r2xx:=0; else r2xx:=∞;
• end
• else begin
        • ryx:=cy/cx;
        • r2xx:=c2x/cx;
• end
• if (cy = 0) then begin
        • if (cx = 0) then rxy:=0; else rxy:= ∞;
        • if (c2y = 0) then r2yy:=0; else r2yy:=∞;
• end
• else begin
        • rxy:=cx/cy;
        • r2yy:=c2y/cy;
• end
• if (|cx| > |cy|) then begin
        • r1:=r2xx; r2:=ryx;
• end
• else begin
        • r1:=r2yy; r2:=rxy;
• end
• ar1:=|r1|; ar2:=|r2|;

• { Search table for best match }
• { ar1[n], ar2[n], a[n], b[n] }
• best_error=∞;
• best_a=0.5; best_b=0.5;
• for each (I:1..n) begin
        • error=(ar1-ar1[i])*(ar1-ar1[i])+(ar2-ar2[i])*(ar2-ar2[i]);
        • if (error < best_error) then begin
                • best_error:=error;
                • best_a:=a[i];
                • best_b:=b[i];
        • end
• end

• { Compute fixed point }
• if (a≥-0.5) then begin
        • pi:=(b+a)/2; pj:=0;
• end
• else begin
        • pi=(2*b-1)/4; pj=(-2*a-1)/(4*(b-a));
• end

• { Calculate reflections }
• if (ar1≠r1 and ar2=r2) best_pj:=-best_pj;
• if (ar1=r1 and ar2=r2) best_pi:=-best_pi;
• if (ar1=r1 and ar2≠r2) begin
        • best_pi=-best_pi;
        • best_pj=-best_pj;
• end
• if (|cy| ≥ |cx|) then
• begin
        • temp:= pi;
        • pi:= pj;
        • pj:=temp;
• end
end
```

Figure 8.4. Pseudo-code for IFT algorithm.

Recall however that the edge is assumed to be within 45° of the vertical, slopes to the right, and lies to the left of the centre of the child block. By considering the symmetry of the problem other orientations can be determined with the same method and look-up table.

Firstly if $|c_1|<|c_2|$ the edge is assumed to be predominantly vertical and the algorithm is identical but the y coefficients and x coefficients are switched. Secondly by using absolute values when searching the look-up table the signs of the ratios can be subsequently used to determine a reflection of the midpoint and give a solution for other cases. If c_3/c_1 is negative and c_2/c_1 positive $\underline{m}$ is reflected in the horizontal axis. If c_3/c_1 is positive and c_2/c_1 is positive $\underline{m}$ is reflected in the vertical axis. If c_3/c_1 is positive and c_2/c_1 negative $\underline{m}$ is reflected in both axes.

Figure 8.4 shows pseudo-code for this algorithm.

8.4 Evaluation

Initial Test

To begin with the method was used to encode a synthetic image, chosen to test a wide range of edge orientations and parent locations. The 256x256 disk image in Figure 8.5 was produced and partitioned into blocks of size 16x16.

Approximations with the bi-quadratic basis, the centred parent, and the implicit fractal parent location were compared and the IFT is seen to dramatically improve the approximation of edges for all orientations and gives a virtually perfect reconstruction of the original.

Practical Application

The implicit fractal technique was then used in the encoding of the test images using an 8x8 block size partition. The PSNR results are shown in Figure 8.6.

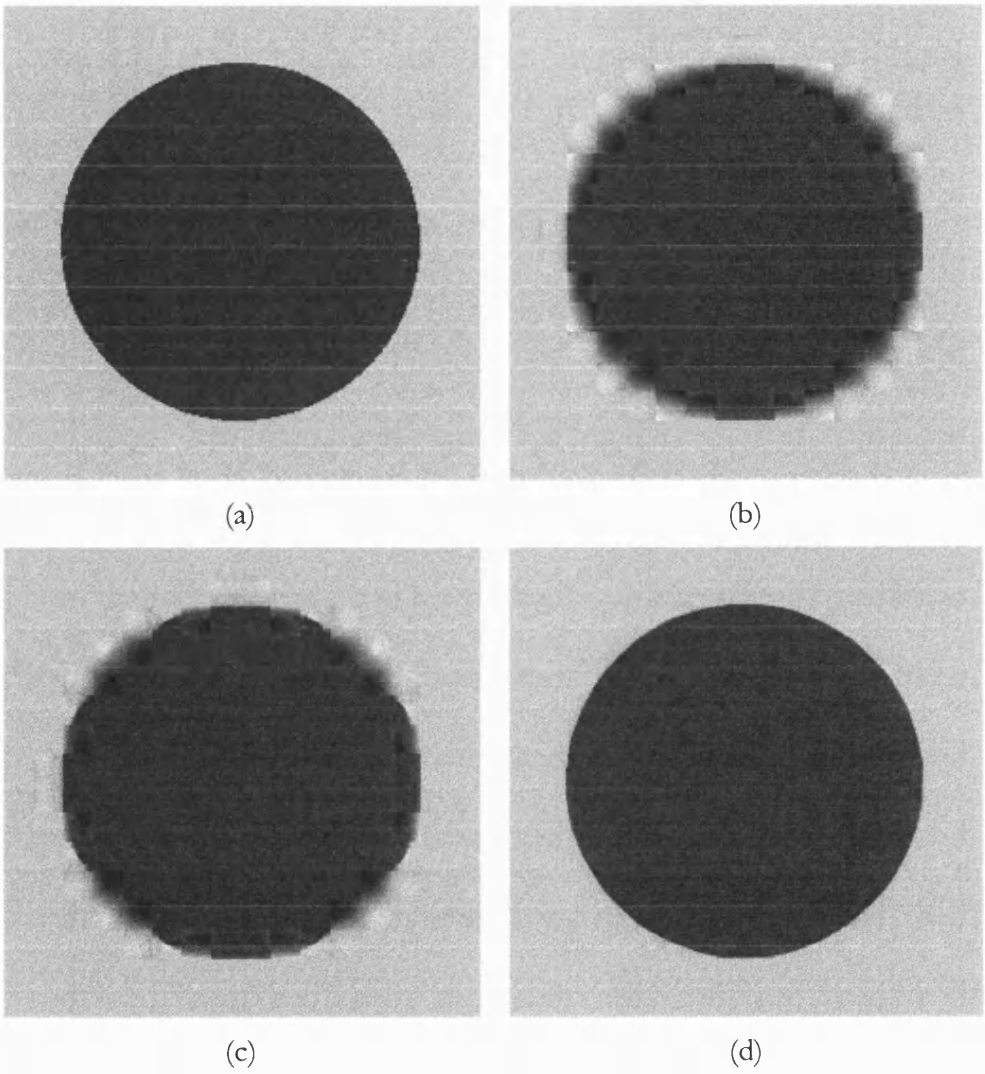


Figure 8.5. Approximation of a synthetic image using 16x16 blocks. (a) Original. (b) Basis approximation. (c) Centred parent. (d) IFT parent.

			Quantized	
Image	Centred	IFT	Centred	IFT
Lena	29.79	30.13	29.34	29.64
Gold Hill	28.99	28.99	28.56	28.59
Barbara	23.68	23.67	23.57	23.56
Boats	27.89	27.90	27.57	27.59

Figure 8.6. Comparative PSNR results for 8x8 block size partition of test images.

It appears the main improvement over the centred case is for the Lena image. Approximation quality in other images does not seem to be affected much by the new technique. Similar results hold when using quantized coefficients, with $Q = 4$.

It appears that the edge model is better suited to some images than others, and when the model does not suit the block the centred parent location is a good choice. Figures 8.7 and 8.8 show the improvement which is obtained by using the method on the Lena image.

Figure 8.7. Fixed block size approximation using centred parent location.

Quadtree Implementation

To evaluate the practical usefulness of the new technique the IFT was implemented in the compression scheme developed over the past three chapters, including use of the rate-distortion switching technique of Chapter 7.

Figure 8.8. Fixed block size approximation using implicit parent location.

On average the new technique resulted in a small increase in fidelity of 0.04dB, a very small improvement, but in fact the method results in an improvement for some images and a loss in fidelity for others. For Lena the average improvement was 0.14dB, for Gold Hill no change, for Barbara 2 -0.07dB, and for Boats 0.12dB. Figure 8.9 shows results for the Lena test image. Visually edges tended to be sharper with the IFT.

8.5 Summary

The implicit fractal technique estimates the optimal parent location well for some blocks and poorly on others. In general it gives performance similar to that of the centred parent but outperforms it on some images.

For images containing a reasonable number of edges similar to that used in the model, e.g. Lena, the IFT gives superior results. In the class of test images

chosen Lena is the only image which is obviously of this nature but the results for Boats were also improved by the IFT. The greatest improvement observed was 0.27dB at 0.1 bits-per-pixel for the Lena image.

Two factors suggest the continued use of this method. Firstly the complexity of the method is low when encoding a block. The length of the lookup table for the 8x8 block size has only 52 entries and the searching process may be performed using integer arithmetic. Secondly the fidelity contributed by greater edge definition is visually more significant than the PSNR results suggest.

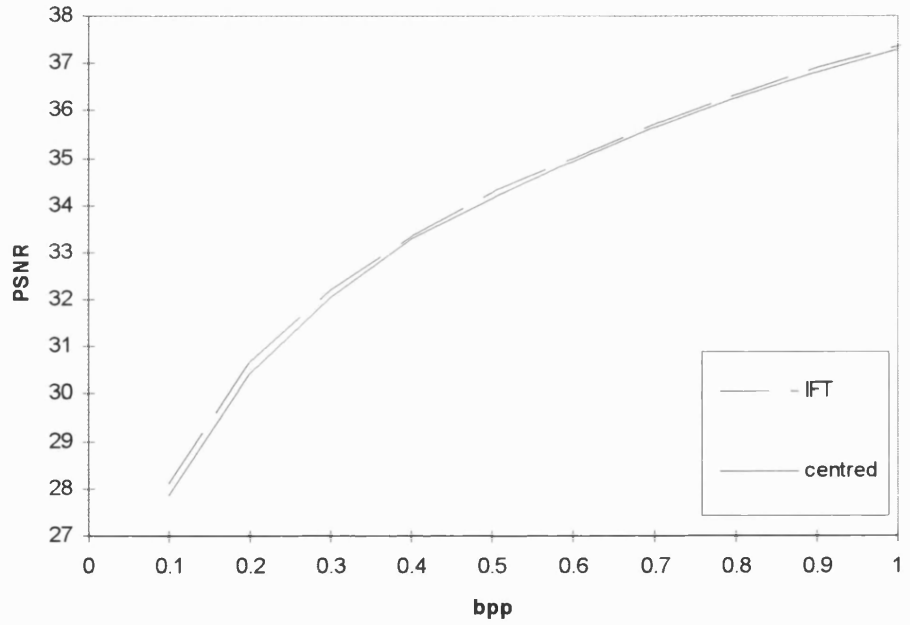


Figure 8.9. Rate-distortion results for Lena test image.

Chapter 9

EXTENSION OF THE IFT

9.1 Introduction

In the previous chapter a novel technique for determining the parent location relative to an image block was presented. In this chapter an extension to this method is examined with the aim of encoding a wider range of image features and hence achieving higher fidelity.

The new method is designed by first examining the usage of basis coefficients by the IFT to see what range of ratios is available that do not conflict with the existing method. This suggests a new edge model and equations are derived in a similar manner to those of the IFT. When the fractal mapping of a block is determined a decision is made, regarding which method to use based on the values of the basis coefficient ratios.

9.2 Preliminary Observations

In the previous chapter the IFT worked because, for a simple edge, the ratios of the basis coefficients have a unique value which can be used to determine the optimal parent. Figure 9.1 shows the distribution of these ratios.

All values are positive for the reasons given previously, but additionally $\left|c_2/c_1\right|$ is restricted to a maximum value of 1 and $\left|c_3/c_1\right|$ has a maximum value of 1.078274.

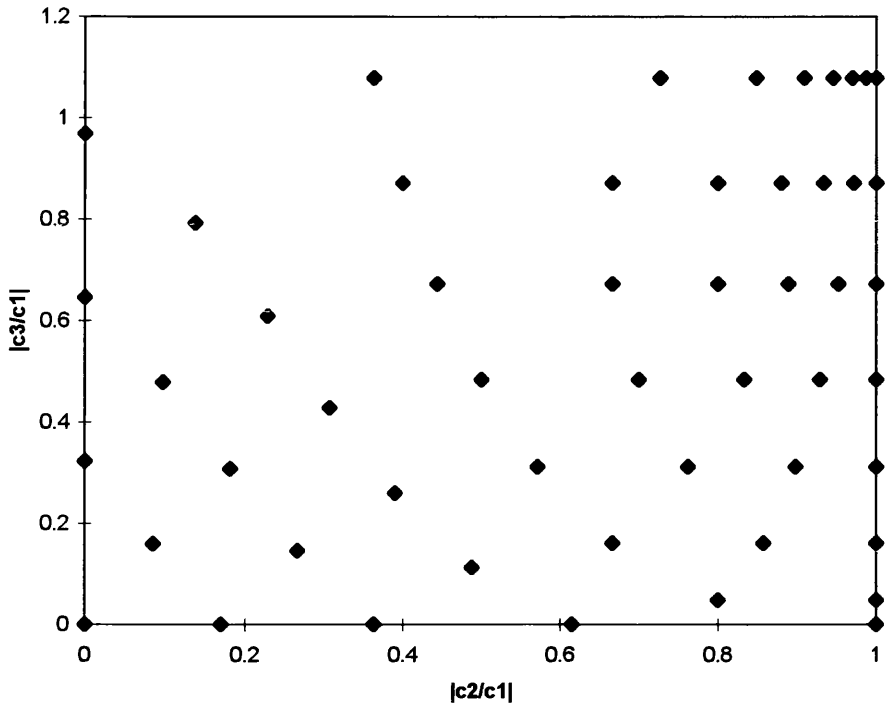


Figure 9.1. Distribution of basis ratios for IFT.

Incidentally, some points are very close together which suggests the values of a and b used to create the table, which simply form a uniform sample, are not optimal. However the non-uniform distribution of the ratios does not appear to affect performance to a significant extent as if this were the case quantization would reduce the fidelity gain of the IFT over the centred parent location, which does not occur. Also, similar ratios correspond to similar block locations and at worse lead to a slight increase in encoding time.

With the range of ratios used by the method restricted, the key to extending the IFT would appear to be the development of a feature model which represents basis functions whose coefficient ratios lie outside this range.

Examining the ratio $\left|c_2/c_1\right|$ in particular, Figure 9.2 shows some sample ratios in one dimension. As the ratio increases past 1 the curve becomes more U-shaped

and it is hypothesised that image blocks producing this effect are the ones which may be approximated by an extended version of the IFT.

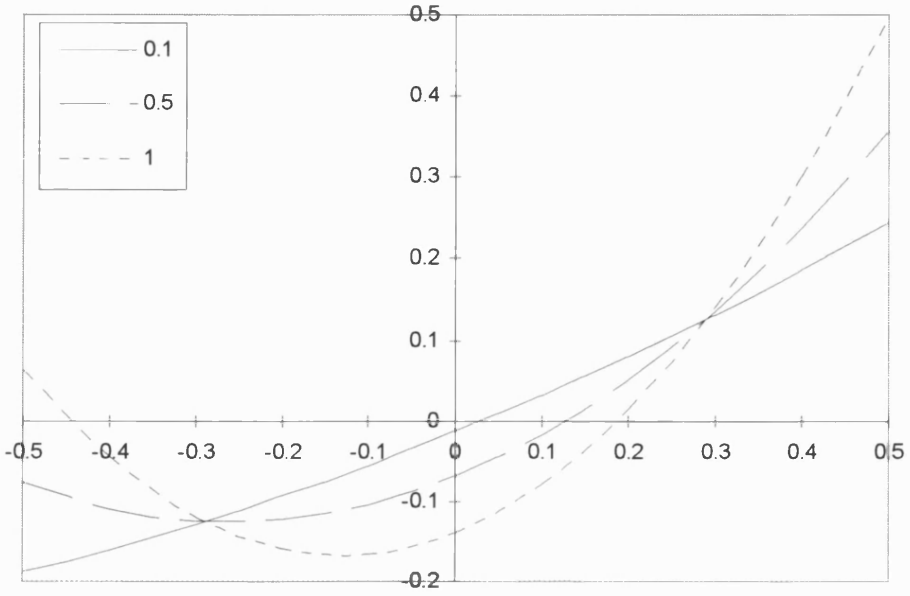


Figure 9.2. Sum of x and x^2 basis functions for different ratio values.

9.3 Dual Edge Model

A simple feature producing $\left|c_2/c_1\right|$ values above 1 is shown in Figure 9.3. This is termed a *Dual Edge Model.*

To approximate this feature a dual parent transform must be adopted. In this transform two parents are designated, one on the midpoint of each line segment. When the fractal mapping is applied the parent with midpoint nearest to each pixel is used for that pixel.

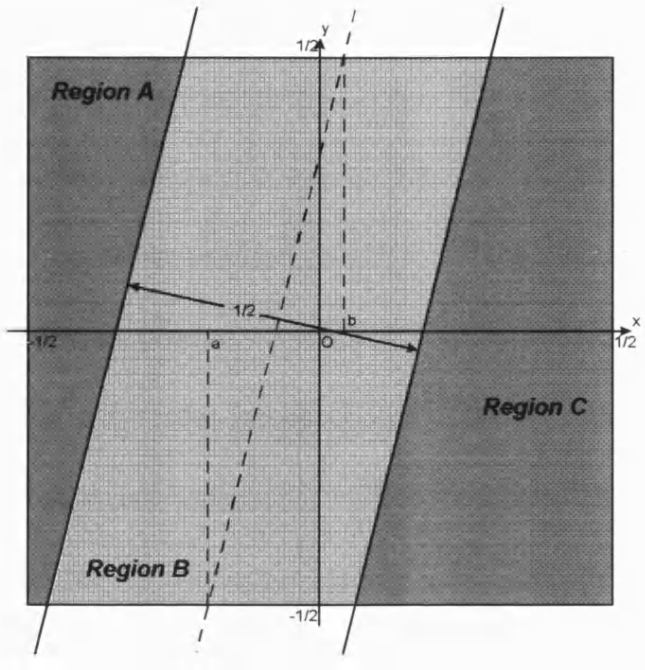


Figure 9.3. Dual edge model.

9.4 Derivation

To derive a solution for this new feature model a similar approach to that taken in Chapter 8 can be adopted. It may be seen that the integral is a sum of two integrals of the type derived for the IFT. In the first integral, regions A and B have intensity value $M/2$ and region C intensity value $O - M/2$. In the second, region A has intensity $O - M/2$ and regions B and C have intensity $M/2$. The sum of these blocks corresponds to the edge model above, and since the sum of the integrals is equal to the integral of the sums the formulae used to derive the IFT can be used again. Details of the solution are given in Appendix C.

9.5 Implementation

A lookup table is constructed as before, from a range of a and b values, using which the parameters of the model and hence the parent locations may be determined from the basis coefficients. The distribution of the resulting ratios

in the lookup table is shown in Figure 9.4. The ratios c_2/c_1 and c_3/c_1 are plotted for each of the five cases in the solution and the absolute values of these ratios were plotted for the IFT.

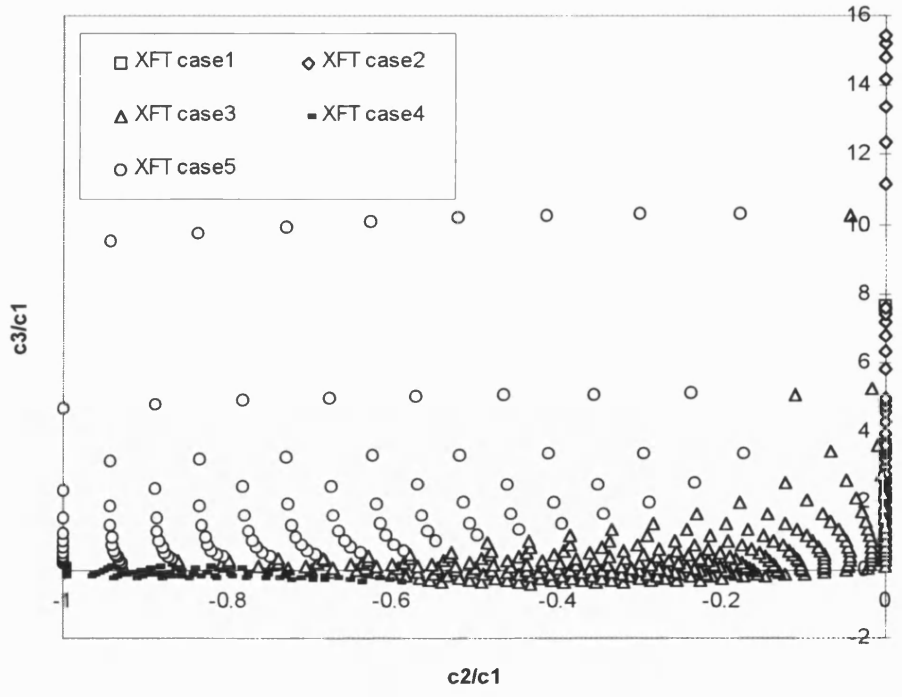


Figure 9.4. Distribution of basis coefficient ratios[1].

As Figure 9.4 shows, when the absolute values of the ratios are stored there will be some conflict with the existing implicit fractal method. However the extended method uses the third ratio c_4/c_1 which does not need to be restricted to positive values. The signs of c_3/c_1 and c_4/c_1 are always identical which means when the absolute value of c_3/c_1 is stored, the negative sign of the

[1] There are a relatively small number of data points for case 1. They are located along the y-axis.

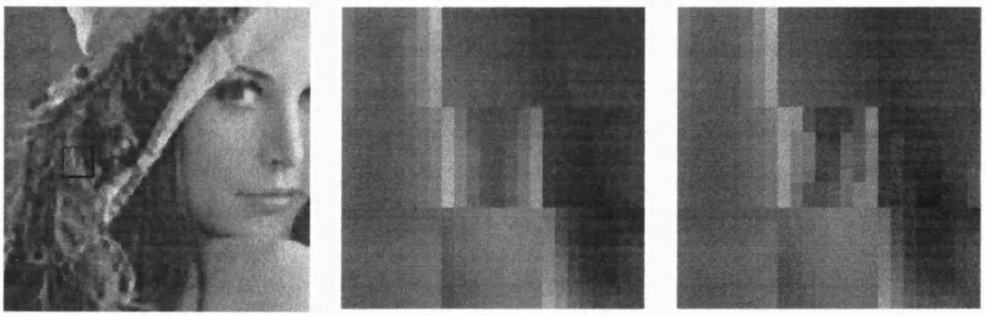
third ratio will still produce good matches. Other cases must be excluded when the magnitude of the first two ratios is within the pre-determined range of the ratios under the IFT and the third ratio is small. Based on empirical results the original edge model is used if

$$\left|c_2/c_1\right| < 1 \text{ and } \left|c_3/c_1\right| < 1.5 \text{ and } \left|c_4/c_1\right| < 1.5 . \tag{9.1}$$

Otherwise the dual edge model is used. New rules for reflecting the midpoints are described in Appendix C.

9.6 Results

Initial tests showed the extended IFT worked successfully, however when compressing practical images very little change in fidelity was produced. For the Lena test image using an 8x8 partition the increase in fidelity of the new method over the original IFT was precisely 0.024608 dB. This very small change corresponded to approximately 10% of blocks using the dual edge method to a significant degree. Figure 9.5 illustrates the kind of improvement seen in this image.



(a) (b) (c)

Figure 9.5. (a) Lena with 8x8 block size approximation. (b) IFT approximation. (c) extended IFT approximation.

When a quadtree is created large blocks tend to use the dual edge model, then as smaller blocks are approximated the single edge model is used more

frequently. Unfortunately it appears the fidelity in the large blocks does not increase sufficiently, due to the dual edge model to give an improvement in the overall performance of the quadtree algorithm. The rate-distortion results showed no improvement for the method.

9.7 Summary

This work on extension of the IFT has been a success in the sense of having created a new implicit fractal method, but a failure in that no gain in coder performance has been achieved. The implicit fractal method introduced in Chapter 8 has been extended in the logical way but with no improvement in the method's effectiveness in terms of visual or numerical fidelity. This suggests the IFT uses the correlation that exists between the fractal and basis parts of the block transform to a practically maximal degree.

Further work in this area could examine the effects of 'normalising' the set of coefficients rather than using ratios. I.e. determining fractal information from the set

$$\left\{ c_1/c, c_2/c, c_3/c, c_4/c \right\} \text{ where } c = \sqrt{c_1^2 + c_2^2 + c_3^2 + c_4^2}\,. \tag{9.2}$$

Also an alternative to the feature model approach is to derive a lookup table from test images, using a statistical correlation found from a large amount of test data. One further area which is potentially rewarding is the use of inter-block correlations.

Chapter 10

PARTITIONING CRITERIA

10.1 Introduction

As explained in Chapter 4 a quadtree partition of the image is created by dividing the image support into 32x32 blocks and recursively splitting blocks with high approximation error into smaller blocks which can be approximated more easily. The key to the success of the process is the estimation of the approximation error in each block.

In [81] several estimation methods were compared and it was found that the amount of edge content, measured by summing the convolution of the image with a Sobel edge filter, was similar in performance to using the approximation error itself. The use of the collage error, the error when mapping the original image onto itself, was shown to perform relatively poorly.

For the reasons given in section 4.4 the collage error has been used so far in this thesis for the purpose of development of fractal transforms, however, now that most of the comparison work has been completed the issue of partitioning criteria can be addressed.

This chapter presents an examination of the effectiveness of several split criteria, including one new method which has not been tried before.

10.2 Methods

To achieve optimal coding performance, the error in a block G should be used as the partitioning criteria, given by the formula

$$\left\| G - \sum_{i=1}^{n} c_i B_i - eH \right\|^2 = \|G\|^2 - 2\sum_{i=1}^{n} c_i \langle G, B_i \rangle + \sum_{i=1}^{n} c_i^2 - 2e\langle G, H \rangle + e^2 \tag{10.1}$$

where H is the parent block from the decoded attractor image.

In practice, because the fractal part of the expression is not known, the error must be estimated. Numerous ways of doing this exist and three of the most likely techniques are described below.

Collage Error

This is the method used in previous chapters and is given by equation 3.2.4. It is related to the collage theorem (equation 3.1.3) and works by replacing H in equation 10.1 with a parent block taken from the original image. By the nature of the coding process this block should be close to the attractor block it replaces.

Edge Value

An edge image is produced by applying Sobel masks, see Figure 10.1, across the image. The edge value of a block is then computed as the sum of the absolute values of the edge image pixels in that block. The idea behind the method is that blocks with more edge information will be harder to approximate and hence have higher approximation errors.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 10.1. Vertical and horizontal Sobel filters.

Pessimistic Collage Error

This is a new method. The collage error uses the original image to predict the attractor error, but in practice many of the details of the image are not present in the attractor image and criterion efficiency is lost. Using equation 3.5.6, a

potentially more realistic error estimate is obtained by averaging parent blocks taken from the original image and the basis approximation, this is termed a *pessimistic collage error.*

10.3 Results

To evaluate the effects of different partitioning criteria a comparison was performed in the usual way. The four test images were compressed at a range of bit rates using the different methods. Figure 10.2 shows the rate distortion curves for the Gold Hill test image which was typical of all four test images. Figure 10.3 shows an example compression of the Lena image at 0.2 bits-per-pixel, with each of the partitioning criteria.

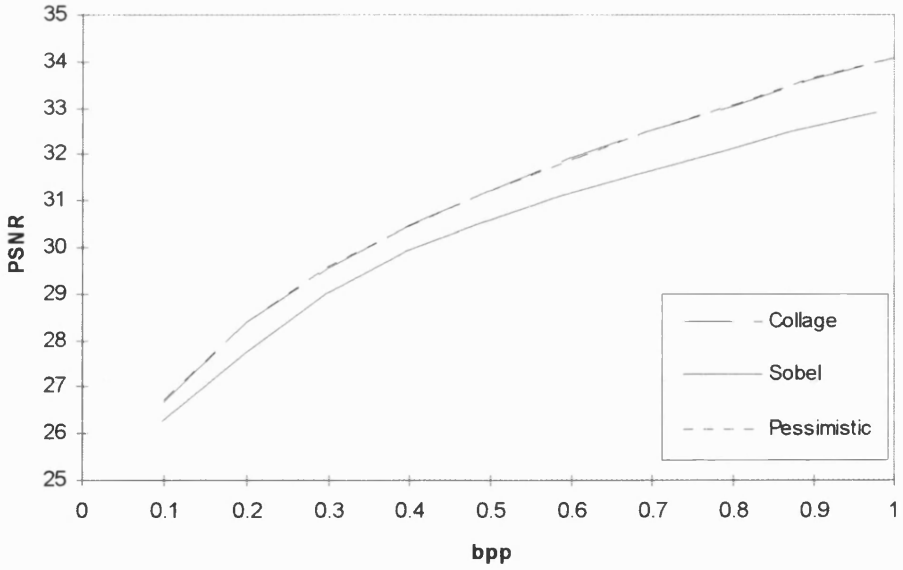


Figure 10.2. Rate-distortion results for Gold Hill test image.

The results show that the previous work of Woolley [81] has been superceeded by new developments in the transform itself. The collage error now

outperforms the Sobel based error by an average of 0.97 dB across the whole range.

The pessimistic collage error performs comparably with the collage error, and produces a similar partition of the image as seen in Figure 10.4.

(a) (b) (c)

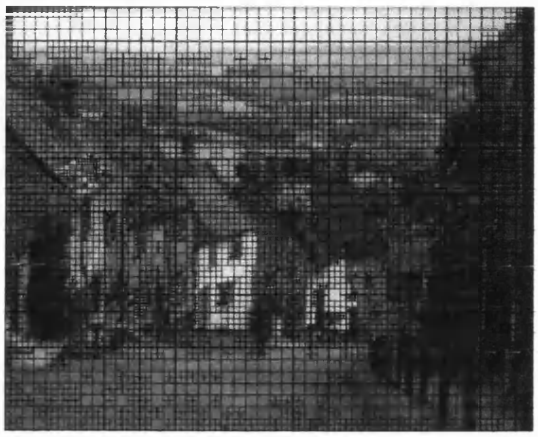
Figure 10.3. Lena image compressed at 0.2 bits-per-pixel, using different partitioning criteria. (a) Collage error. (b) Sobel error. (c) Pessimistic collage error.

10.4 Summary

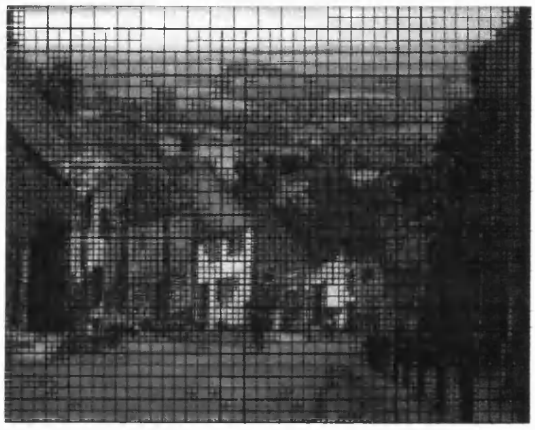
In this chapter three quadtree partitioning criteria were examined, two of which were examined previously in [81] and a third which was introduced in this chapter.

As a result of developments in the fractal transform the collage error is now the best partitioning method of these three.

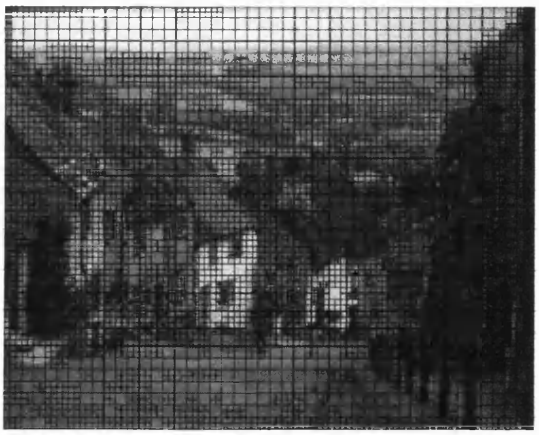
These results additionally justify the earlier argument, given in Chapter 4, for the use of the collage error rather than Sobel based error.



(a)



(b)



(c)

Figure 10.4. Partitions of the Gold Hill test image based on different criteria. (a) Collage error. (b) Sobel error. (c) Pessimistic collage error.

Chapter 11

PROPOSAL: A NEW FRACTAL TRANSFORM

11.1 Introduction

One limitation which applies to fractal image compression is the maximum possible fidelity the block transform is capable of achieving. With almost all fractal transforms the fractal part of the approximation is a contracted, scaled, and possibly rotated and reflected block from elsewhere in the image. Some work has used more advanced mappings based on variations of the above scheme to improve fidelity [7].

In this chapter the notion of local self-similarity is defined formally and this definition is used to generalise fractal coding to give it the potential for further improvement. An innovative new technique is presented which is capable of giving far greater fidelity and is particularly remarkable in that it is independent of the image partition used.

11.2 Definition of Local Self-Similarity

The self-similarity used by fractal image compression is commonly referred to as *piecewise self-similarity*. An image is piecewise self-similar if there exist regions of the image, usually rectangular blocks, which are identical or near-identical under affine transformation. *Local self-similarity* is a special case where one region is contained in a larger piecewise self-similar region, such as in the IFT and BFT.

This later case is important as previous studies [81,80] have shown it to be the most useful type of self-similarity for image compression. For local self-similarity the affine transformation has a unique fixed point within the child

block. For example, in the IFT this fixed point is the midpoint of a line segment. Since under the affine mapping all other points are mapped closer to the fixed point, this point is referred to as the *centre of self-similarity*, and is illustrated in Figure 11.1.

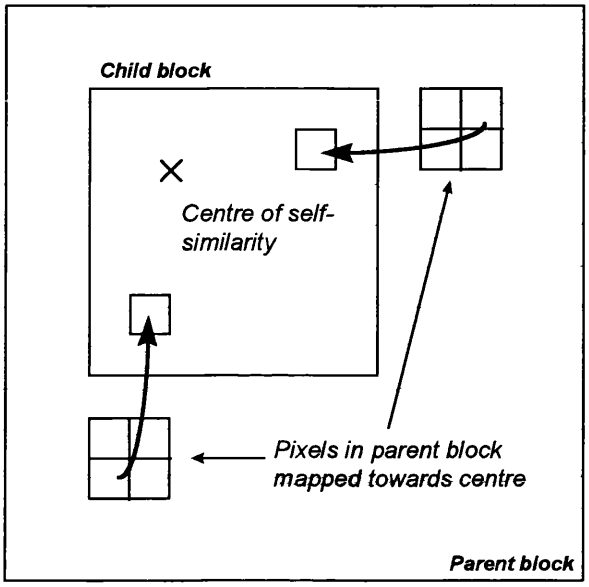


Figure 11.1. Local self-similarity.

11.3 Theory

Using the above description, a new fractal transform is defined based on the use of centre points of self-similarity, or *seed points*, where each seed point will represent the centre of self-similarity in some local region. These seed points are used to define a *self-similarity function*, $W(\mathbf{x})$, which is equivalent to the union of the contraction mappings normally used to describe the mapping of parent blocks onto child blocks. This function describes which pixels of the image are mapped onto which pixels at the decoding stage, and so explicitly describes the self-similarity of the image. A fundamental aspect of encoding an image with this transform will be the choice of the number of seed points and their location. No theoretical approach to this aspect of the transform is described here, but the problem is discussed in later sections.

Definition

Let $\{\mathbf{x}_j; j = 1...N\}$ be a set of seed points and S the image support, and let the image support be partitioned into cells $\{S_j; j = 1...N\}$ such that each cell contains only one seed point.

Then the self-similarity function $W(\mathbf{x})$ is defined by

$$W(\mathbf{x}) = \mathbf{x} - \left(\frac{1}{c} - 1\right)(\mathbf{x}_j - \mathbf{x}) \tag{11.2}$$

where $\mathbf{x}_j$ is the seed point in the cell containing $\mathbf{x}$, and c is the ratio of the contraction to be used.

It should be clear that if the partition of the image support consists of child blocks and the ratio of contraction is 2, then $W(\mathbf{x})$ describes the usual mapping between parent and child. For example, in Figure 11.1, the centre of self-similarity is a seed point and the child block is one cell of the partition.

Voronoi Partitioning

The partition used by this method is independent of the usual quadtree partitioning of the image and so must be determined by some other means. One possible solution is to use an automatically computed *Voronoi partition.*

By associating pixels with the nearest seed point a Voronoi partition of the image support is created with each cell one region of self-similarity. A mapping is thereby created for the whole image where each pixel is mapped onto by pixels from a location determined by the nearest seed point. Domain and range blocks used in traditional fractal image compression are replaced and a more powerful tool for capturing the self-similarity of images is produced.

For each seed point $\mathbf{x}_j$ there is one cell, denoted S_j, the set of points in S which are closer to the seed point than any other. I.e.

$$S_j = \left\{ \begin{matrix} \mathbf{x} \in S; d(\mathbf{x},\mathbf{x}_j) < d(\mathbf{x},\mathbf{x}_i) \forall i < j, \\ d(\mathbf{x},\mathbf{x}_j) \leq d(\mathbf{x},\mathbf{x}_i) \forall i > j \end{matrix} \right\} \tag{11.1}$$

The result is a partition of the image support illustrated by the Voronoi diagram in Figure 11.2. This mapping is independent of the quadtree image partition used and is effectively block-less.

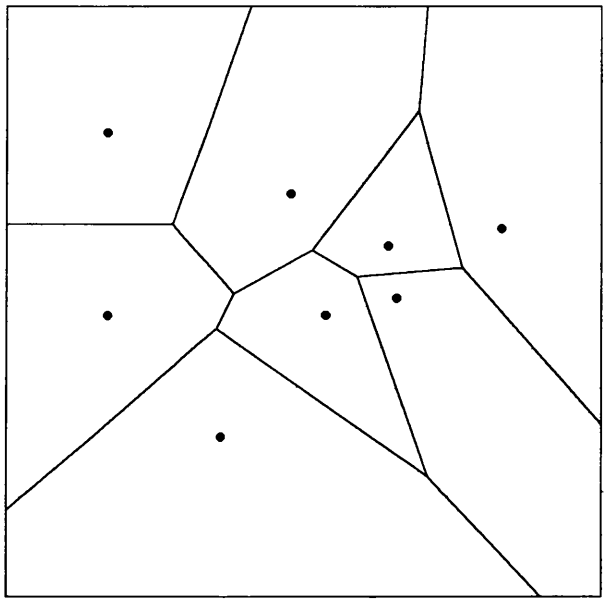


Figure 11.2. Voronoi diagram resulting from eight seed points.

Decoding

The decoding of the image is achieved as with traditional fractal compression schemes, except that the function W is used in place of w_k^{-1} in equation (3.2.1).

11.4 Implementation

This transform may be implemented in a variety of ways but to illustrate its potential a sample image was encoded using seed points placed along the edges of the word "Fractal" in the image and a Voronoi partition was computed.

The decoded image showed an impressive level of fidelity compared to conventional methods, as seen in Figure 11.3

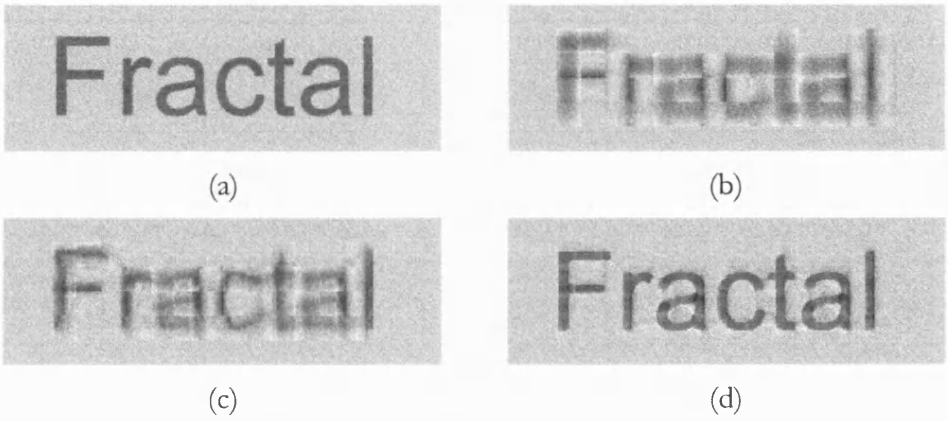


Figure 11.3. Effects of new fractal transform. (a) Original image. (b) Basis. (c) Local searching fractal approximation. (d) New seed point-based approximation.

11.5 Further Work

The description of self-similarity in this method is block-less when used with a Voronoi partition. The key problems which must now be solved are the determination of the seed points, and their storage - both of which may be computationally costly.

The selection of seed points is the main challenge. Initially a set of equally spaced seed points could be chosen and then additional points added in regions which have high predicted approximation error - analogous to the block-based quadtree method used in previous chapters.

An aim of further work could be to remove blocks from fractal coding altogether. By choosing, say, 1000 seed points a set of 1000 areas is created which could replace the block based partitions used in traditional methods altogether. The approximation of the original image intensity in each region could be achieved using modified versions of current methods, with a basis defined on a support which was specific to each block. This may cause further

difficulties however when quantizing the resulting coefficients unless the basis functions are normalised for every different region shape.

The most difficult problem remains the determination of new seed points, as the addition of seed points changes the surrounding regions rather than just the one which the seed point lies in. This means a large amount of computation is required to try all or any significant number of possible seed point locations. A simple method will need to be found which can predict the best location of the seed point. Perhaps an IFT-like method would succeed.

11.6 Summary

In fractal coding, a function w_k traditionally describes the similarity of one block to another. It has been shown that the use of rotation and reflection is inefficient and that local searching is particularly effective. In these cases the seed point of the mapping completely describes the self-similarity of the image for these two blocks, assuming that the ratio of contraction is fixed.

This new transform comprises a set of seed points representing centres of self-similarity in given areas which are used to define a self-similarity function W for the image. The flexibility of the transform makes the improvement of local searching methods possible. Each point is treated individually and the fractal part of the transform is block-less. Many current methods are subsets of this method.

A new fractal transform has been proposed which has the potential to increase the fidelity of fractal coding. Little investigation of how it may be implemented has been presented. The provisional results obtained from the synthetic image indicate it will be of value when used correctly in a finished compression scheme. However, much work will be needed to determine an efficient means of identifying and storing seed points, and working with polygonal partitions. One possibility is to form an initial set of seed points and anneal or use a

genetic algorithm to improve. Such a solution would, however, be computationally very costly.

Chapter 12

RESOLUTION ENHANCEMENT AND ZOOMING

12.1 Introduction

The increasing of image resolution may be desired in several situations. For example when viewing a photograph a user may wish to zoom-in on an area of interest, or when an image is printed at poster size it needs to be interpolated, or an image may be taken from a low-resolution image source, such as many CCD cameras, and can be viewed more easily if the resolution is first increased.

As explained in Chapter 2, fractal codings are independent of resolution, describing a surface over a continuous support. To decode at a finite resolution iterates are produced using appropriate decimation. Following this approach it should be clear that the image may be decoded to a higher resolution than the original and by the nature of the self-similarity mappings the new image has detail at finer scales not previously present.

An image may be decoded at a higher or lower resolution than the original quite easily, and by extension individual areas of the image may be *zoomed.* I.e. displayed independently at a higher resolution. The problem of both zooming and resolution enhancement is that the extra detail created may not correspond well to the original. The new detail may give a better quality image than before, but this is not guaranteed. Since the original is not available at the new resolution, the results can only be evaluated subjectively, or synthetically by sub-sampling before zooming.

Previous investigations have been carried out in this area, including [19], [26] and [52]. This chapter examines several aspects of the problem: the differences between common fractal transforms when zooming, zooming on compressed

images, the maximum performance of fractal transforms for zooming, and the enhancement of fractal codes for improved results.

12.2 Implementation

To decode at a different resolution to that of the original is not difficult. Assuming the fractal code stored in memory contains block locations measured in pixels, these values must be scaled by the appropriate factor and, critically, the basis and fractal coefficients must also be scaled by the same factor because of the normalisation process (see Chapter 5).

To zoom on one part of an image implies determining a higher resolution decoding of the image in that section. Exactly what this involves depends on the fractal transform used. As parent blocks are mapped from areas in the image beyond the section of interest, a larger area may need to be decoded, dependent on the number of iterations used. If searching is part of the transform, and blocks are mapped from non-local parts of the image, then the whole image will typically need to be decoded at the higher resolution. Transforms using local self-similarity, such as the IFT and the centred parent BFT, have a computational advantage in that only a relatively small area surrounding the image section of interest needs to be decoded.

12.3 Results

Comparison of Fractal Transforms

A number of versions of the block transform have been developed and the main differences are the use of searching and the complexity of the basis. The four test images were encoded with three different fractal transforms designed to be representative of those commonly in use. The first is based on Jacquin [35], involving only local searching with a DC basis. The second uses the BFT bi-quadratic basis, also with local searching. The third is the implicit fractal technique presented earlier.

In the zoomed images a sharpening of features was observed, and if the approximation at the original resolution was good the high resolution image was better. Some extra distortion was observed when the approximation quality was lower, particularly when using the local searching transform with DC basis. Figure 12.1 shows a x16 zoom of the brim of Lena's hat which illustrates this effect.

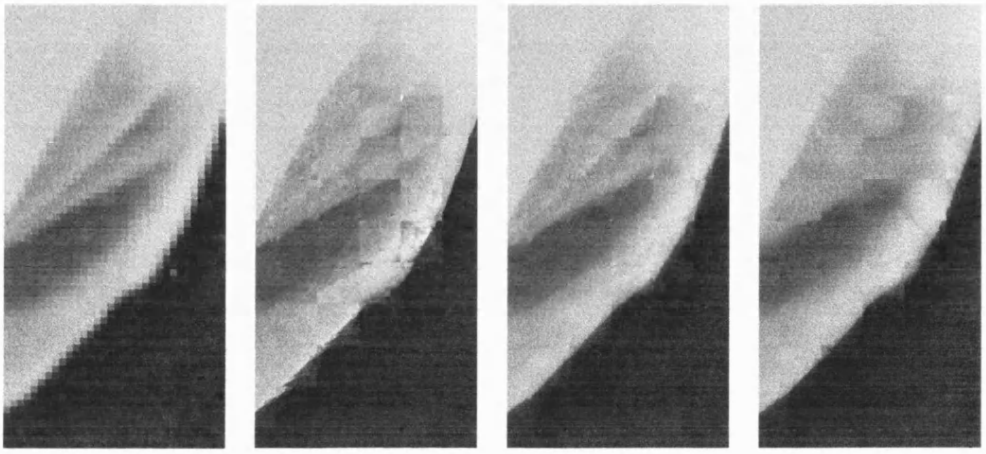


Figure 12.1. Lena hat, zoomed approximations. (a) Original displayed at same size. (b) Jacquin-type fractal transform with DC basis. (c) BFT with bi-quadratic basis and local searching. (d) Implicit fractal technique.

Zooming Compressed Images

In practice a fractal coding will result from compressing an image and this gives a quadtree partition with quantization error as well as detail loss caused by the transform.

These extra factors have a visible effect on the zoomed image quality, but only in proportion to the effect on the approximation at the original resolution. Blockiness and loss of detail are still apparent, but when the original approximation is good the zoomed image has sharpened features. Figure 12.2 illustrates this, showing a zoom of Lena's hat at 0.2 bits-per-pixel. For

comparison the approximation is also shown with resolution increased by an anti-aliasing filter, which is commonly used for this purpose.

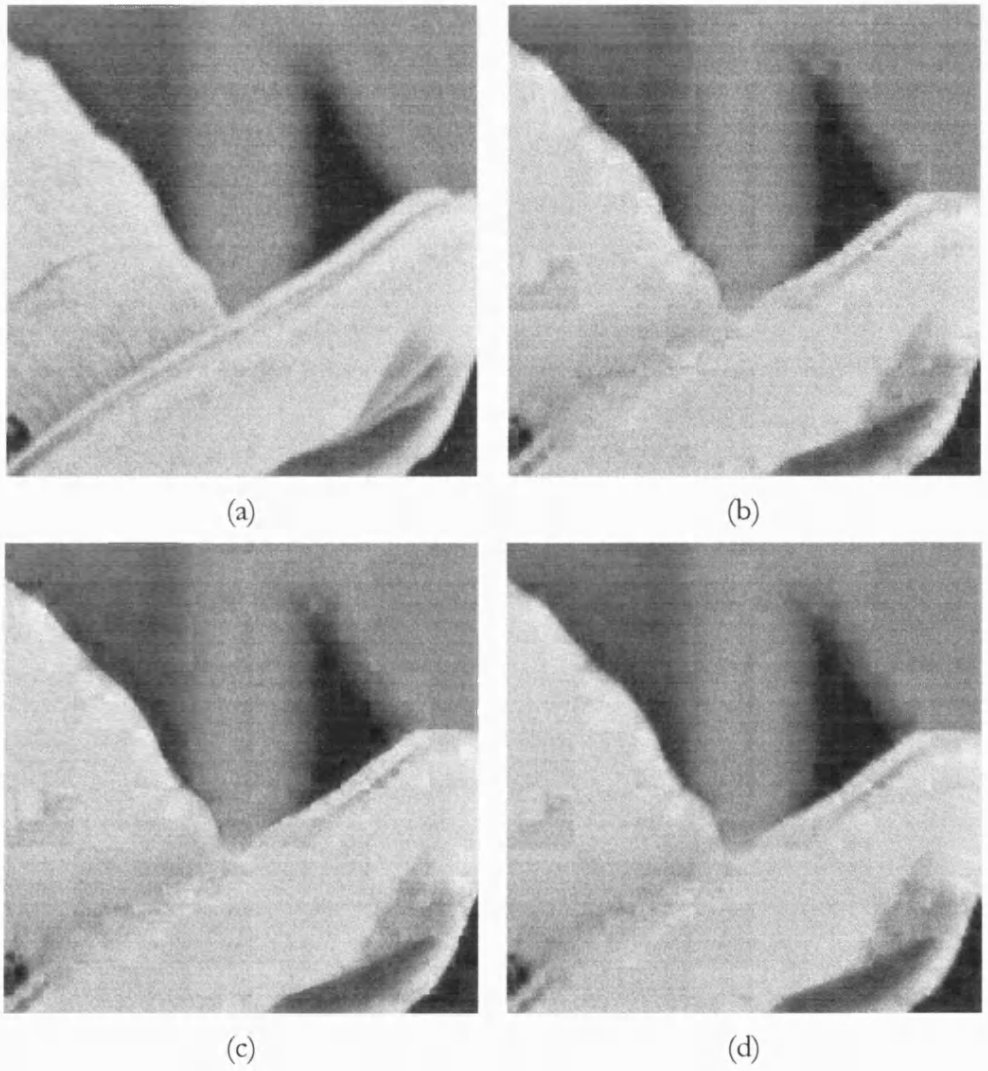
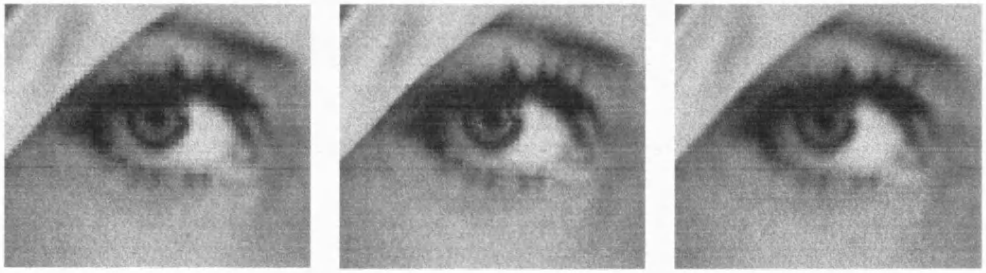


Figure 12.2. Lena compressed at 0.2 bits-per-pixel. (a) Original image. (b) IFT approximation. (c) Fractal zoom. (d) Interpolated approximation.

High Fidelity Zooming

The maximum effect achievable when zooming a fractal encoding is of interest as the fractal transform could potentially provide a novel method of enhancing resolution in uncompressed images. To achieve maximum fidelity images were encoded using a 4x4 block size, local searching transform, with bi-polynomial

basis. Again features were sharpened when the images were zoomed, although a small amount of visually annoying artefact is introduced. Comparison with the conventionally interpolated image favours the fractal zoom, see Figure 12.3.



(a) (b) (c)

Figure 12.3. Zoomed high fidelity approximation. (a) Original image. (b) Zoomed fractal approximation. (c) Fractal approximation interpolated by anti-aliasing filter.

12.4 Enhanced IFT Zooming

It has been shown that the performance of fractal coding schemes may be improved by using implicit parent locations because of their very low bit-cost. As a result the IFT is given some special consideration here. The result of zooming with an implicit fractal transform can be seen in Figures 12.1 and 12.2. However the optimal use for compression of this transform means only a relatively small number of blocks use a fractal term. When zooming this can be more noticeable. For example the approximation of Lena's shoulder in Figure 12.4(b) shows one block in the top left does not use a fractal term despite being part of an edge. Whereas further down the shoulder, artefact is created where a non-fractal block produces a small 'notch' which is propagated by mapping onto neighbouring blocks.

The implicit fractal zoom may be improved by introducing new fractal terms into the fractal code. Using the mathematical edge model from the coding stage the intersection of edges in fractal blocks with non-fractal blocks is estimated. If two edges meet the sides of a non-fractal block, the block is assumed to have

an edge in the original judged to be too insignificant in the MSE sense to be coded or an edge which was lost when the image was digitised. The parent location and fractal coefficient is then computed based on the location of the intersecting edges and the average of the coefficients of the fractal blocks. The result is the elimination of this kind of artefact, see Figure 12.4(c).

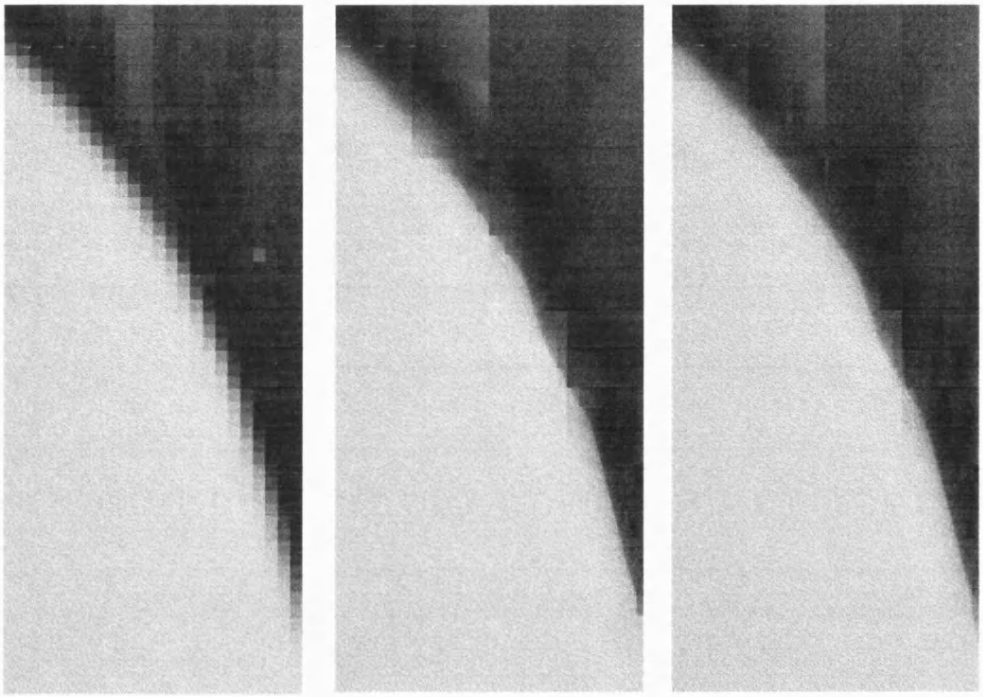


Figure 12.4. Enhanced IFT zooming. (a) Original image section. (b) Zoomed implicit fractal approximation. (c) Enhanced zoom.

12.5 Summary

From the results in this chapter several observations can be made:

- When edges and other image features are well approximated at the original resolution they are sharp and fairly well preserved in the zoomed or resolution enhanced images

- The DC basis transform results in more distortion than other transforms, while the more complex basis, as used in the BFT and IFT, gives more stable decoding in both the local searching transform and the implicit fractal cases

- Edges do not always match well at block boundaries even when the approximation quality at the original resolution is good

Zoomed images were produced from several different fractal coding techniques commonly employed for compression. The zooming process worked as expected from the theory and the fractal zooms produced looked generally better than the original approximations. The fractal transform gave sharp edges in each of the edge blocks preserved in the approximation at normal resolution, although edges between blocks did not always line up. The performance of the zoomed implicit fractal transform was improved by introducing additional fractal terms.

A further observation is that fractal coding techniques can be used to increase the resolution of uncompressed images. Using a small block size partition, a complex basis, and searching locally for the best parent-child mapping the image can be coded at a very high fidelity then decoded at a higher resolution. The result is an image which has sharper features than if a more traditional interpolation method was applied.

Chapter 13

POST-PROCESSING

13.1 Introduction

There are three types of artefact present in images compressed with fractal methods, namely loss of detail within blocks, quantization error, and blockiness. Little can be done about the detail lost within a block but the quantization error and blockiness can be quite easily reduced after the image is decoded by introducing a post-processing stage where the image is filtered. The result can be a significant improvement in visual quality especially at higher compression ratios.

Relatively little work has been carried out on filtering fractal coded images in this way, see [40] and [19]. Common filters such as median and averaging filters may be applied, and can be used across the image or only at block edges. New methods could be created by first modelling the block error and then deriving a filter.

This chapter presents a comparison of filtering methods when applied to images compressed with the fractal transform developed in this thesis.

13.2 Methods

Median Filtering

The median filter is widely used in image processing, for example to remove impulse and other noise. To filter a pixel all pixels within a certain range are examined and the median pixel value is found. This is a relatively complex method in computational terms and increases in complexity exponentially with the filter size. Filters are of size 3x3, 5x5, 7x7 etc.

Averaging Filter

The averaging filter takes all pixel values with a certain range and averages their values. Wherever the averaging filter is applied the image is smoothed. Filters are of the same sizes as the median filters.

Weighted Averaging Method

This is a common method, suggested for fractal image compression by Fisher [19], and is applied only at block edges. If a and b are adjacent pixels from different child blocks, these are replaced by $w_1 a + w_2 b$ and $w_2 a + w_1 b$ respectively. For blocks of size 4x4 $w_1 = 5/6$ and $w_2 = 1/6$. For larger blocks $w_1 = 2/3$ and $w_2 = 1/3$. For 2x2 blocks no averaging is used. For larger blocks of size 16x16 or greater the degree of blockiness is increased and pixels internal to the block are smoothed as well. If the pixel values are a, b, c, and d, with b and c being the two adjacent pixels from different blocks, their values are replaced by $\frac{3a+2b+c}{6}$, $\frac{2b+c}{3}$, $\frac{b+2c}{3}$, and $\frac{b+2c+3d}{6}$ respectively.

13.3 Results

To evaluate the different methods an average performance was computed for each method for each block size over the four test images. The 3x3 averaging and median filters were applied both for every pixel and only at block edges. The results are shown in Figure 13.1.

block size	median filter at block edges	averaging filter at block edges	Fisher's method	median filter	averaging filter
4x4	-2.32	-2.80	-0.21	-2.73	-3.40
8x8	+0.05	0.00	+0.06	+0.02	-0.14
16x16	+0.11	+0.16	+0.12	+0.14	+0.18
32x32	+0.04	+0.10	+0.11	+0.07	+0.13

Figure 13.1. Improvement over original PSNR for methods with 3x3 filter.

No filter improves the results for 4x4 block size, and the 16x16 block size shows the greatest improvement from filtering. Fisher's method, described earlier, generally performs well, as shown in Figure 13.2.

Fisher's method
-0.21
+0.06
+0.12
+0.11

Figure 13.2. Improvement over original PSNR for Fisher's method.

For the larger block sizes the discontinuity between the blocks is more apparent and greater filtering may be justified. To determine whether this was the case 5x5 filters were applied in the same manner. The results, in Figure 13.3 show that the improvement can be greater, though not for the 8x8 block size. For the 16x16 block size the median filter performs slightly better. The averaging filter performs worse than before. At the 32x32 block size all methods give improved results.

block size	median filter at block edges	averaging filter at block edges	median filter	averaging filter
8x8	-0.20	-0.49	-0.48	-1.11
16x16	+0.14	+0.13	+0.16	+0.07
32x32	+0.08	+0.11	+0.13	+0.17

Figure 13.3. Improvement over original PSNR for filtering methods with 5x5 filter.

Fisher's method is simple and, when the 4x4 block size is excluded, gives competitive results. Consequently it may be regarded as the best method.

To determine the effects of this filter in practice the four test images were compressed at rates of 0.1 to 1.0 bits-per-pixel. After post-processing the average improvement was 0.11dB. Figure 13.4 shows the Lena image

compressed at 0.2 bits-per-pixel before and after post-processing[1]. The effects of averaging and median filtering block edges is also shown.

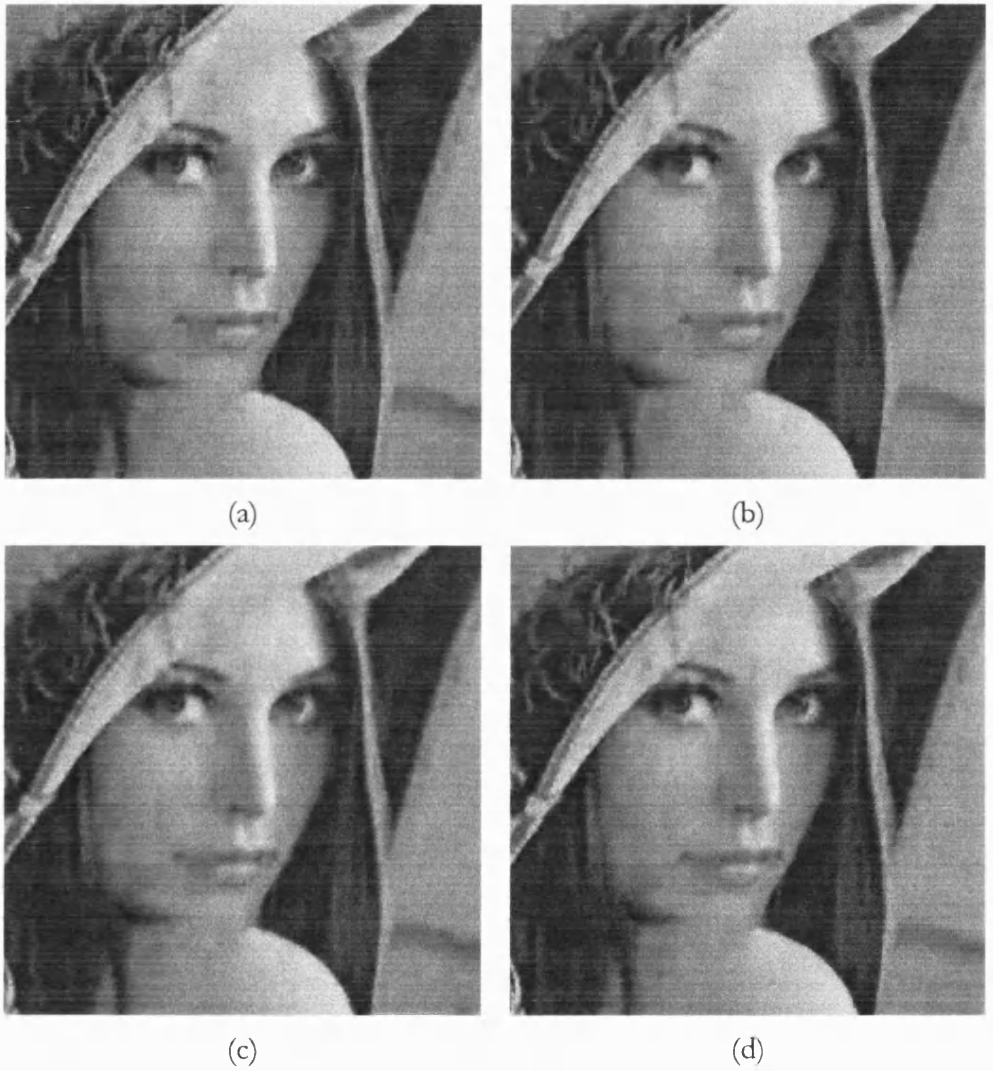


Figure 13.4. Comparison of post-processing methods. (a) Lena compressed at 0.2 bits-per-pixel, PSNR 30.68dB. (b) Median filtered image, PSNR 30.78dB. (c) Averaging filtered image, PSNR 30.80dB. (d) Image filtered with Fisher's method, PSNR 30.98dB.

[1] Blockiness is less apparent when the image is printed than when it is displayed on a computer screen.

13.4 Psycho-Visual Trial

To evaluate the visual quality of the filtering methods, two sets of images were produced and volunteers asked to choose the best from each set. A 128x128 section of the Gold Hill image was coded with 8x8 blocks and a 256x256 section of the Lena image was coded with 16x16 blocks. Each image was then filtered with the methods listed in Figures 13.1 and 13.2. Figure 13.5 shows the results.

Filtering method	Gold Hill		Lena	
	PSNR	Rating	PSNR	Rating
median filter at block edges	24.14	7	24.33	3
averaging filter at block edges	24.15	5	24.39	4
Fisher's method	24.32	7	24.36	9
median filter	24.06	2	24.38	3
averaging filter	24.04	10	24.46	13

Figure 13.5. Psycho-visual results, showing PSNR and number of viewers who thought image was best. Unfiltered PSNR was 24.31dB for Gold Hill image section, 24.16dB for Lena image section.

13.5 Summary

In this chapter several basic filtering methods were examined and their suitability for post-processing images compressed with the IFT was evaluated. The conclusion was that the best method was that of Fisher [19] which gave good PSNR performance compared to other methods and is computationally very simple.

The result of post-processing with this transform is an average increase in PSNR of 0.11dB, and visually most of the blocking artefact is eliminated. More improvement occurs for highly compressed images, where blockiness is greater.

A psycho-visual trial suggested that PSNR and observed quality were not highly correlated. In particular the PSNR for the Gold Hill image section with the averaging filter applied was the lowest in the group, however it was the most popular with viewers. For both images Fisher's method and the averaging filter scored well. The median filter was particularly unpopular.

When zooming images blocking artefact is also a significant problem and no attempt to solve it has been undertaken here. The filters considered are too 'weak' to be effective in this case, since the block sizes involved are much larger.

Chapter 14

SUMMARY

Background Work

This thesis began by describing some of the basic ideas behind fractal image compression and proceeded to survey the research work done in this area.

Jacquin's original method represented a breakthrough as the first fully automatic fractal image compression scheme, and much work followed its publication. Most sought to improve and extend the method, but more recently hybrids have been created using other image compression techniques. A wide range of ideas have been published for improving the determination of transform coefficients, modifying the transform itself, and improving the way domain blocks are mapped to range blocks. Attempts have been made to determine the optimal quantization and entropy coding methods and the best partition scheme (quadtree, HV, triangular, or polygonal).

The searching part of Jacquin's algorithm has received special attention. Being slow to encode, a wide range of schemes for accelerating the process using classification, hierarchical searching, and fast domain-range match computation have been proposed and investigated.

Concerning the decoding stage, pyramid decoding schemes have been proposed, and analysis of alternative criteria for convergence has been carried out. Also the effects of zooming and increasing image resolution with fractal transforms have been examined.

The BFT added to the body of fractal image compression research by extending the massic part of Jacquin's transform. Much work has been done to find the optimal design parameters, including the degree of searching, number

of massic parameters, use of isometries, and partitioning methods. Previous work has shown the best implementation to be that of the non-searching, bi-quadratic case, without isometries, using a quadtree partition with only light quantization of the coefficients.

As pointed out by Wohlberg [80], the design of fractal transforms presents a number of problems, since there are few theoretical results on which design decisions can be based. Chapter 4 described the approach used to develop fractal compression throughout the remainder of the work, based on the comparative performance of compression schemes over four test images at bit rates of between 0.1 and 1.0 bits-per-pixel.

The theory behind the fractal compression schemes in this work was laid out in Chapter 3, this chapter described in detail how an image may be specified as a fixed point of a contraction mapping, termed a *fractal image operator*, operating on a space of images. The Collage Theorem provides a mechanism for determining the contraction mapping in terms of a number of parameters, and these parameters are then stored in place of the original image.

To decode, a sequence of images is constructed by applying the fractal image operator repeatedly to any initial image, and it was noted that this procedure may be extended to allow *zooming* on images and resolution enhancement.

Two possible modifications to the basic fractal transform were also explained, namely *orthogonalisation* and *normalisation*. The first of these decorrelates the coefficients of the basis and parent blocks, and the second normalises them to simplify the encoding process.

Finally, Chapter 3 described the method of linear quantization of transform coefficients and analysed the effects of different degrees of quantization.

Chapter 4 described the focus of this work in more detail. The BFT is one competitive method of fractal image compression and it was decided that this

transform was a promising subject for further development and could also be used as a basis for new transforms. The state of the art, as regards the BFT, was therefore reviewed in detail, referring to the work of Monro, Dudbridge, Woolley, and Bethel.

The conclusions of Woolley that a non-searching, bi-quadratic transform with quad-tree partition is the optimal implementation was adopted, although it is acknowledged that the use of more advanced techniques may potentially alter Woolley's results. Uniform quantization followed by Huffman coding was adopted for use throughout the thesis based on its relative simplicity, good performance, and that it allows fair comparison of different implementations.

However, the quad-tree partitioning criteria comparison carried out by Woolley was effectively disregarded. The choice of Sobel based criterion is counterproductive for the purposes of fractal transform development - where fractal mappings are intended to adequately describe edges and texture in blocks.

A method of comparative evaluation for new work was also given, as mentioned above.

Improvements to the BFT

Firstly Chapter 5 studied the use of orthogonal and orthonormal transforms. The main result was that both types of transform are better than the original, for several reasons. From fixed block size results, the basic potential of an orthogonal transform for approximation is greater, and when implemented in a quadtree scheme the comparative fidelity increased slightly further, suggesting that the orthogonal transform may be better suited to the collage based split criterion. From the quantization study it was clear that the orthogonal transform is also significantly less susceptible to quantization error than the non-orthogonal equivalent; orthogonalised fractal coefficients may be

quantized 25 times more heavily. The conclusion is that orthogonal transforms give significantly better overall results for image compression.

The use of a normalisation factor in the parent mapping was investigated and showed a practically identical level of performance to the orthogonal transform.

Chapter 6 presented an investigation of the parameters used to quantize the fractal coefficients. Unlike most previous studies several test images were used and this gives the results an increased value. By comparing the rate-distortion performance of the fractal compression scheme under different parameter values it was seen that the optimal choice varies from image to image and between bit rates. A method was subsequently introduced which successfully determined the parameter value automatically. Using a formula derived from the previous results, optimal performance of the method was obtained consistently over all test images and bit-rates.

The next topic of investigation was the rate-distortion based switching technique invented by Bethel, which uses a fractal term in a block if and only if it improves the overall rate-distortion performance of the scheme. It was found that several variations on the method were possible and that the optimal implementation carries out switching after the partitioning of the image is complete. The method used in [77] did not give the best results. Better results were obtained when fractal terms were initially used for every block.

Using the rate-distortion switching technique saves significantly on the bit cost. At 0.4 bits-per-pixel an average of 28% more blocks are available to the partitioning process. The average area of the image approximated using fractal terms at this bit-rate is 18% and as expected blocks using fractal terms are those with more complex details such as edge blocks and texture blocks.

New Techniques

The implicit fractal technique is a new invention which estimates the optimal parent location from the basis coefficients in each block. Compared with the centred parent scheme used previously the IFT can give noticeable improvement in image fidelity, particularly for images like Lena and Boats. On other images it gives performance similar to that of the centred parent.

The method works by modelling edges in terms of several parameters. A straight edge with constant grey level values either side is assumed to run through a block, and formulae for basis coefficients can be derived in terms of the location of the line. The optimal parent location is easily computed for a simple edge, and a look-up table containing sets of basis coefficient ratios with corresponding parent locations can be created. Using this table in both encoding and decoding stages allows a parent location to be determined for each block from just the basis coefficients by searching the look-up table for a closest match.

This is a predictive method and requires no extra bits to specify the parent location. On average the method gives an improvement in fidelity, particularly in certain types of image, to which the edge model is well suited.

Two advantages to the method are its low complexity: the length of the lookup table for the 8x8 block size has only 52 entries and the searching process may be performed using integer arithmetic. And secondly the fidelity contributed by greater edge definition is visually more significant than the PSNR results suggest.

In Chapter 9 the IFT was extended to use a more advanced edge model and so improve fidelity in each image block. The ratios of coefficients used by the existing technique were examined to determine what range of values were unused by the method. A new *dual edge model* was then created which produced ratio values generally distinct from the ratios of the IFT. From the model,

formulae for coefficients, and hence a look-up table, were derived as before. A switch was then implemented in the block encoding and decoding functions which would determine which edge model to use.

The extended IFT worked in approximating a wider range of blocks with the fractal part of the code, but the number of blocks which benefited from the dual edge model was sufficiently small to leave the overall rate-distortion results unimproved. It was concluded that the IFT uses the correlation that exists between the fractal and basis parts of the block transform to a practically maximal degree.

Chapter 10 examined three quadtree partitioning criteria using the fractal transform developed over previous chapters. Two of these were examined in [81]. Comparing performance over the full set of test images showed that the collage error was now the best choice for partitioning, as a result of developments in the fractal transform. A novel criterion was tried but did not improve results.

Chapter 11 introduced a new fractal transform. In fractal coding an affine map usually describes the similarity of one block to another. But when rotations and reflections are not used, local self-similarity can be conveniently described using a *seed point.*

A new transform is described which uses seed points to represent centres of self-similarity in given areas, making the improvement of local searching methods possible. Each point is treated individually and the fractal part of the transform is block-less. Many current methods are subsets of this method. The new fractal transform has the potential to increase the fidelity of fractal coding although little investigation of how it may be implemented has been done. Provisional results obtained from the synthetic image indicated it will be of value when used correctly in a finished compression scheme.

Decoding

Resolution enhancement and zooming on a particular area of an image are two intrinsic advantages of fractal image coding. Relatively little work has been carried out to examine the effectiveness of fractal transforms for this purpose and Chapter 12 attempted to redress the balance.

After experimentation several observations were made:

- When edges and other image features are well approximated at the original resolution they are sharp and fairly well preserved in the zoomed or resolution enhanced images

- The DC basis transform results in more distortion than other transforms, while the more complex basis, as used in the BFT and IFT, gives more stable decoding in both the local searching transform and the implicit fractal cases

- Edges do not always match well at block boundaries even when the approximation quality at the original resolution is good

Zoomed images were produced from several different fractal coding techniques commonly employed for compression. The zooming process worked as expected from the theory and the fractal zooms produced looked generally better than the original approximations. The fractal transform gave sharp edges in each of the edge blocks preserved in the approximation at normal resolution, although edges between blocks did not always line up.

To improve the results from zooming images compressed with the IFT a method of introducing additional fractal terms was designed. This used the intersection points of edges in adjacent blocks with the block boundary to predict an edge location. The parent was then placed so that the midpoint of

the edge was the common point of both the child and parent. The fractal coefficient was the average of the fractal coefficients of the adjacent blocks.

It was also found that fractal coding techniques can be used to increase the resolution of uncompressed images. Using a small block size partition, a complex basis, and searching locally for the best parent-child mapping the image can be coded at a very high fidelity then decoded at a higher resolution. The result is an image which has sharper features than if a more traditional interpolation method was applied.

The post-processing of images to remove compression artefact is common in block-based coding schemes. Several basic filtering methods were compared in Chapter 13 when post-processing images compressed with the IFT. It was found that blockiness could effectively be reduced by an averaging filter applied at block edges. The best method was that of Fisher [19] which gave good performance compared to other methods and is computationally very simple. The average increase in PSNR is 0.11dB with this method, and visually most of the blocking artefact is eliminated. More improvement occurs for highly compressed images, where blockiness is greater.

Conclusion

Overall this thesis has presented work which includes a sequence of improvements to fractal compression, beginning with the Bath Fractal Transform and ending with an orthogonalised implicit fractal transform, with optimal quantization, rate-distortion switching, improved partitioning and effective post-filtering. The increase in fidelity over the original method is considerable, as illustrated in the next chapter.

Chapter 15

CONCLUSIONS

Discussion

A number of studies and new ideas were presented in this thesis, each designed to improve the fidelity obtained by fractal transforms at any given bit-rate and for any image.

The first of these was orthogonalisation, which gives a significant improvement in the performance of the BFT. Previously it had been applied to a Jacquin-type compressor with the intention of reducing complexity, but in this work it was shown that its real benefit lies in improvement of the quantization of the transform coefficients. With greater quantization of the fractal coefficient the performance of the method in rate-distortion terms is improved.

The study of quantization led to the development of a novel automatic quantization scheme which adapted the coarseness of the quantization according to the complexity of the image and the desired degree of compression. This optimal result is advanced compared with other work done in the fractal image coding field, including that of Fisher and others, and previous work on the BFT. It is a step forwards in making fractal image compression schemes more competitive. Further study may prove useful - more complex formulae derived from a wider range of images may give better results.

Rate-distortion switching for fractal image compression was proposed by Bethel and improved here. This method gives a useful improvement in the fidelity at any given bit rate by using fractal terms optimally. The results in this thesis show this to be an excellent improvement which can be made to any

fractal compression scheme which creates its partition progressively. Probably best performance is obtained with schemes using a more complex basis where the basis part of the transform is more likely to give an adequate approximation on its own.

The implicit fractal technique introduced in this thesis is a novel and quite advanced addition to the fractal image compression literature. It was first published in [77] and is a particularly good example of technique development. Beginning with the conjecture that a correlation exists between basis coefficients and parent location, the problem is modelled, and from this model a solution which is practically applicable is derived. Results for the IFT show an improvement in fidelity over the centred parent case, which is often particularly striking at edges in the image.

The basic idea of correlation between the fractal and non-fractal parts of the scheme may be more widely applicable, however. Any scheme may in principle be improved by using a fractal mapping if a modelling of image features is carried out. For example, the effects of a straight edge can be modelled in terms of the parameters of a compression scheme and from this a correlation may potentially be found and used to predict fractal information.

The IFT was extended to use a dual edge model as well as the normal model, but it was concluded that useful extension of the IFT from its existing form is probably impossible. The study presented showed that a range of basis ratios was unused by the IFT, and the most common image feature corresponding to these ratios was chosen. However no significant improvement in fidelity was possible. The information contained in the basis coefficients is presumably too limited.

A study of quadtree partitioning showed that the collage error is the best partitioning criterion. A better approach to this problem may be to examine the estimation of the error on a block wise basis to gain a more detailed insight

in to the workings of different methods. The accuracy of the collage approximation could therefore be evaluated exactly, and this could provide further insight into the workings of the fractal transform.

A new fractal transform has been proposed and is intended as a step towards realising the original intention of using fractals to compress images. The novel feature of this new transform is that the fractal relationships are not block based and are more flexible than with conventional fractal coding. Much further work is required to produce an effective compression method with this transform but this could prove a rewarding area of study.

Zooming and resolution enhancement were examined in this work and the main conclusion was that if the original approximation quality is good the zoomed image should be visually superior. The key problem is that small errors in the approximation are also magnified to a large scale when the image is zoomed. A method of increasing resolution in uncompressed images was also examined. Further work will show whether this is a useful method, by comparison over a larger number of images.

Blockiness resulting from large blocks in the image partition is visually quite noticeable and post-processing methods were examined to try to reduce or eliminate this effect. Of the limited range of options compared it was concluded that Fisher's method gave best results and removed most of the blocking artefact without destroying detail.

Comparative Performance

This thesis began by adopting the BFT in its optimal implementation, as known in 1996 from the publication of [81]. Improvements have been made and new transforms introduced which give greater fidelity at a given bit-rate and the difference in performance is illustrated by Figure 15.1 which shows 'before' and 'after' rate-distortion curves.

JPEG remains the standard image compression method, although JPEG 2000 is under consideration, and the rate-distortion performance of JPEG is also shown in Figure 15.1. The IFT outperforms JPEG at high compression ratios but gives inferior performance at lower ratios. The main reason for this is that this fractal transform has been developed from the BFT which has always been designed as a low complexity method. In particular it has only a small number of parameters to be computed per block, and does not usually use searching. From Figure 15.1 it may be seen that the point at which the fractal scheme becomes more efficient than JPEG has risen to 0.36 bpp for the Lena image, from 0.23 bpp, which is quite useful.

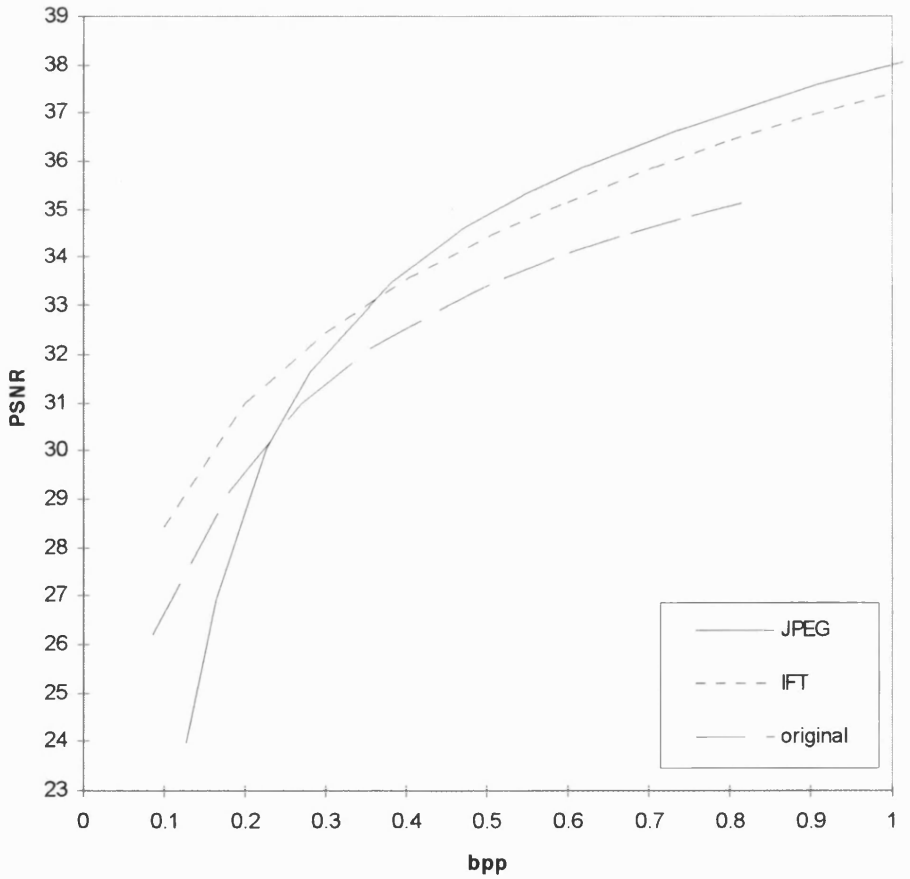


Figure 15.1. Comparison of rate-distortion performance over Lena test image.

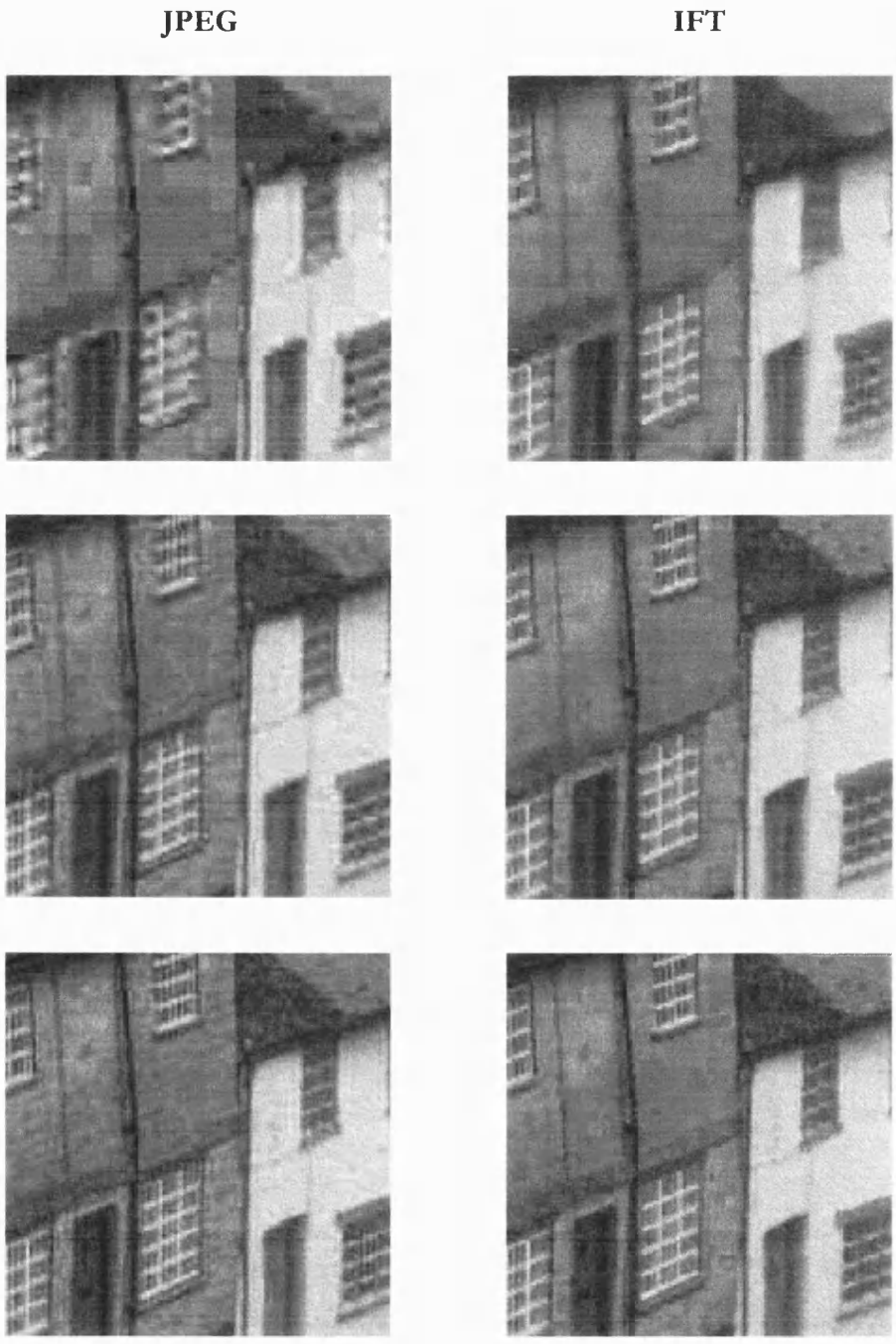


Figure 15.2. Comparison of JPEG compressed images with IFT compressed images, at rates of 0.2, 0.4, and 0.6 bits-per-pixel. PSNR increase from using the IFT is 1.40dB, 0.00dB, and -0.38dB respectively.

Future Work

The difference in performance between fractal coding methods and other methods, such as wavelet coding, has narrowed so that a relatively small further improvement would make fractal coding a truly competitive compression technology.

To improve block-based fractal coding further there are several avenues of investigation. The use of more complex bases or fractal mappings is one area, but also the details of the schemes introduced in this thesis may be investigated in more detail.

One important area of research which has received little attention is the nature of self-similarity, though [80] is one study. The assumption of piece-wise self-similarity is made in every fractal coding scheme, but it is important to understand what degree of self-similarity to expect. After studying this problem, Wohlberg has concluded that a small but significant degree of local self-similarity is to be expected, but questions whether this implies that fractal image coding is worthwhile.

This leads to the question of whether further work on block-based fractal coding is indeed justified. Despite the extensive work presented in this thesis fractal coding is still not competitive with wavelet coders. At best it may be said that fractal coding is of low complexity and gives very good performance at low bit-rates - two characteristics which make it suitable for numerous applications, for example multimedia. To increase the performance at lower compression ratios a larger basis or fractal mapping incorporating multiple fractal terms may be effective.

Hybrid schemes such as fractal-wavelet coding show greater promise. Several researchers [16,38,79] have shown that fractal coding may be re-formulated in a wavelet context. This leads to an improvement in compression and decompression time, increased compression performance, and removes the

tiling artefact of block based fractal coding. An application of the IFT concept in this area of research may prove worthwhile.

Conclusion

This thesis has attempted to always use *fractal* methods to achieve better results. An overall improvement may always be obtained by hybridising with a better image compression method where the fractal component plays are relatively minor role, but this does not lead to a better understanding of the problem of exploiting self-similarity.

This thesis continued the development of fractal block transforms and has achieved improvements in fidelity with several novel techniques. The resulting transform has the advantage of very competitive performance at low bit-rates with low computational complexity.

Appendix A

PUBLISHED WORK

Here is a list of published work which I have co-authored. Papers marked with an asterisk are reproduced in the following pages.

Conference Papers

* 'Hybrid image compression with implicit fractal terms', P. D. Wakefield, D. M. Bethel and D. M. Monro, *Proceedings of International Conference on Acoustics, Speech and Signal Processing '97*, pp. IV-2933-2936.

'An implicit fractal transform', D. M. Monro, D. M. Bethel and P. D. Wakefield, *Proc. Advanced Digital Video Compression Engineering, July 1997.*

* 'Zooming with implicit fractals', D. M. Monro and P. D. Wakefield, *Proceedings of International Conference on Image Processing '97*, pp. I-913-916.

* 'Fractal enhancement of decompressed images', P. D. Wakefield and D. M. Monro, *Proceedings of International Conference on Image Processing '98.*

'Image compression with implicit fractals', P. D. Wakefield and D. M. Monro, *Proceedings of Fractal 98.*

Patents

P. D. Wakefield and D. M. Monro / News Digital Systems, 'Improvements in or relating to encoding a portion of a digital image' (Implicit fractal technique). (*Applied for*).

HYBRID IMAGE COMPRESSION WITH IMPLICIT FRACTAL TERMS

P. D. Wakefield, D. M. Bethel and D. M. Monro

School of Electronic and Electrical Engineering, University of Bath
Claverton Down, BA2 7AY, England

e-mail: paulw@ee.bath.ac.uk daveb@ee.bath.ac.uk D.M.Monro@bath.ac.uk
Fax: +44 1225 826073
Internet: http://dmsun4.bath.ac.uk/

ABSTRACT

The performance of any block based image coder can be improved by applying fractal terms to selected blocks. Two novel methods are used to achieve this. Firstly the coder determines whether a local fractal term will improve each image block by examining its rate/distortion contribution, so that only beneficial fractal terms are used. Secondly, the decoder deduces the offset parameters for the local fractal transform from the basis functions alone, by inferring the dominant edge position, so that no offset information is required. To illustrate the method, we use a quadtree decomposed image with a truncated DCT basis. Using a standard test image, the proportion of the picture area enhanced by fractals decreases from 16.1% at 0.6 bpp to 8.1% at a high compression ratio of 80:1 (0.1bpp). The fractal terms contribute less than 5% of the compressed code in all cases. The PSNR is improved slightly, and edge detail is visually enhanced.

1. BACKGROUND

To compress an image, define an Iterated Function System (IFS)[1-5] of order N to be $W = \{w_k; k = 1,..., N\}$, where the w_k are contraction mappings, each defined on a subset A_k of the image support. The attractor of W is a non-overlapping tiling of the image, as in Figure 1. A fractal function $f(x, y)$, is then defined which approximates the brightness $g(x, y)$. An image block taken from the location A_k is referred to as the parent and an image block taken from $w_k(A_k)$ is referred to as the child. For each tile the function is specified by a recursive mapping v_k such that

$$f(w_k(x, y)) = v_k(x, y, f(x, y)) \quad \text{for } (x, y) \text{ in } A_k. \qquad (1$$

In this work we use mappings of the form

$$v_k(x, y, f) = p_k(x_0 + \delta x,\ y_0 + \delta y) + e_k \tilde{f}_k(\delta x, \delta y) \qquad (2$$

where (x_0, y_0) is the bottom left corner of A_k and $p_k(x_0 + \delta x,\ y_0 + \delta y) = \sum_{i=1}^{n} c_i\, b_i(\delta x, \delta y)$, is an approximation by basis functions $\{b_i\}$, e_k is the single fractal coefficient

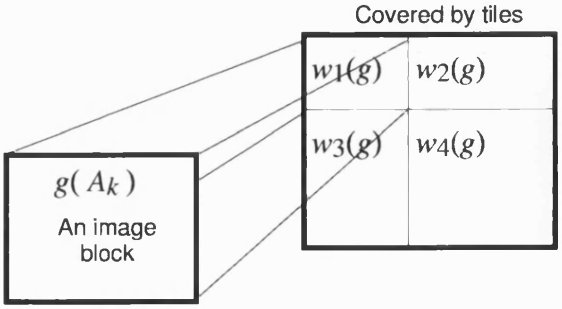


Figure 1. Fractal transforms apply contraction mappings of parent blocks onto child blocks.

and $\tilde{f}_k$ is the parent block $f_k(\delta x, \delta y) = f(x_0 + \delta x, y_0 + \delta y)$ orthogonalized with respect to the basis using

$$\tilde{f}_k = f_k - \sum_{i=1}^{n} \langle b_i, f_k \rangle b_i \qquad (3$$

where $$\langle b_i, f_k \rangle = \iint_{A_k} f_k(x, y)\, b_i(x, y)\, dx\, dy \qquad (4$$

and the basis functions are normalized by $\langle b_i, b_i \rangle = 1$

To solve, the known image $g(x, y)$ is used in place of the unknown fractal function $f(x, y)$ and the approximation is known to be valid by the Collage Theorem [1].

The process is fractal because of the self-similarity inherent in v_k. The mappings form an ensemble of functions which, when iterated or otherwise rendered [5], form an approximation to the image. Usually the tiling of the image is by square or rectangular child blocks, and it is often assumed that p_k is a simple brightness level. Much work has concentrated on reducing the complexity of searching for the best parent to map onto each child [6, 7].

An alternative approach uses more complex basis functions [7, 8] and restricts or even eliminates searching. Such an approach is the Bath Fractal Transform (BFT) [9, 10] with which a pre-determined tiling without searching gives the greatest accuracy at a given compression ratio, when used with a quadratic basis. In combination with the Accurate Fractal Rendering Algorithm (AFRA) [5], the BFT has been used for real time fractal video [11].

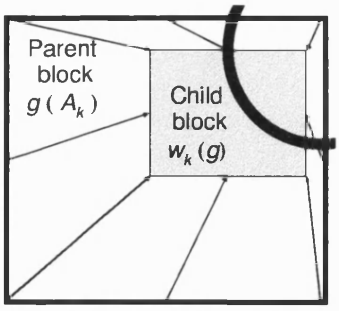


Figure 2. A local fractal transform, with the child block inside the parent block. For a strong edge, the parent/child relationship can be deduced from the basis function.

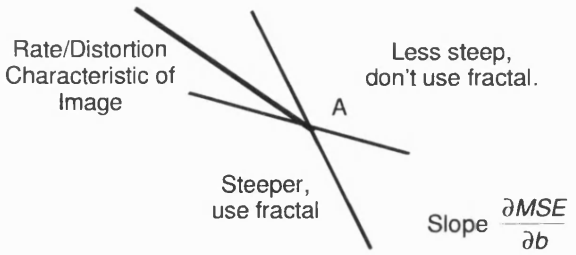


Figure 3. A block is selected for fractal enhancement by the terminal slope of the rate/distortion characteristic.

2. IMPLICIT FRACTALS

In this work we improve the performance of non-searching fractal compression by allowing any relative position of child block and parent block provided the child is local to the parent. We start from the observation that fractal terms enhance edges in images provided the alignment of the edges in the mapping is correct. To exploit this, we determine the dominant edge of the child block from the basis functions alone, using a mathematical model of the edge. We position the parent block so that the edge passes through it in the same relative position, as illustrated by Figure 2. The method applies to any choice of basis functions and to any image partition.

Once the coder has calculated the basis coefficients for a child block, the same process of edge determination and matching to find the parent can be carried out as will be used in the decoder. The coder can then decide whether using a fractal term will improve the rate/distortion characteristic of the image. The compressed code contains only the fractal coefficient e_k, because the offset of the child within the parent will be computed by the decoder from the same information that was used by the coder.

The coder froms a non-fractal approximation to all or part of the image by any coding method. The code for any image block adds Δb bits to the total image code, and increases its total MSE by ΔMSE. If the slope of the rate distortion curve is determined numerically by the coder as it compresses a partition of the image, then the terminal slope $\partial MSE/\partial b$ is know quite accurately, point A in Figure 3. The coder can then examine each image block to determine whether a fractal term will improve the compression. It is assumed that the terminal slope will not be altered appreciably by this process.

If a fractal is used, it will contribute further bits, Δb_{frac} and improve the image MSE by ΔMSE_{frac}. The fractal term is beneficial in rate/distortion terms if and only if

$$\frac{\Delta MSE_{frac}}{\Delta b_{frac}} < \frac{\partial MSE}{\partial b} \qquad (5$$

(i.e. is more negative at point A in Figure 3.)

as illustrated by Figure 3. The estimate of Δb_{frac} can be completely accurate, even if entropy coding is used. The collage theorem [1, 2] would normally be used to estimate ΔMSE_{frac}, which is not completely accurate although bounds on its accuracy can be computed. The gradient is also more difficult to compute if the basis functions and the fractal component are not orthogonal.

3. IMPLEMENTATION AND EVALUATION

To evaluate the method, for $p_k(x, y)$ we used a DCT basis limited to the 6 terms C_{00}, C_{01}, C_{02}, C_{10}, C_{11} and C_{20}. $C_{00}$was quantized to 7 bits, and the other coefficients to 6 bits, with a fixed Huffman table derived from a test set of images. An image is partitioned into 32 x 32 pixel blocks, and within each partition a quadtree structure was formed so that the basis approximation MSE was distributed as evenly as possible over the image. Because it seldom occurs that a fractal term is used with the initial partition, it is acceptable to carry out the fractal selection only as blocks are split.

Given the DCT coefficients, we can classify a block as being predominantly horizontal or predominantly vertical by comparing $|C_{10}|$ to $|C_{01}|$. To apply the implicit fractal, we search a 2D table using the edge model, which gives the edge location as a function of the ratios

$$r_{xx} = \left|\frac{C_{20}}{C_{10}}\right| \text{ and } r_{yx} = \left|\frac{C_{01}}{C_{10}}\right| \text{ for a vertical edge and}$$

similarly $r_{yy} = \left|\frac{C_{02}}{C_{01}}\right|$ and $r_{xy} = \left|\frac{C_{10}}{C_{01}}\right|$ for horizontal.

It can be shown that these ratios are independent of the intensity on either side of the edge according to the model used, Figure 4. The location is reflected horizontally and/or vertically according to the signs of the ratios.

The fractal coefficient is then found by solving equation 2 for e_k. By the Collage Theorem, $f(x, y)$ is approximated by the original parent block $g(x, y)$, orthogonalized with respect to the basis [1, 7]. In our experiments, using a

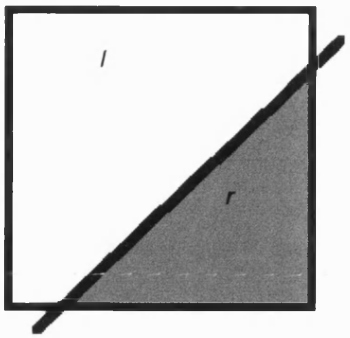


Figure 4. A simple model of an edge, in which *l* and *r* are the flat intensities on either side of the edge.

Huffman coded fractal coefficient e_k adds less than 2 bits to the code of each block in which it is used. Because the decoder can determine the fractal offset vectors from the basis approximation, the fractal terms are very efficiently coded.

Figure 5 shows the improvement in edge definition that can be obtained in a synthetic image. The implicit fractal method improves all orientations of the edge by computing the optimum parent location from the decoded basis function.

In Table 1 we list the results of coding the test image Gold Hill over a range of compression ratios. Figure 6 shows clearly the improvement in edge detail obtained in the Lenna image where the implicit fractal enhancement is applied.

4. DISCUSSION AND CONCLUSIONS

We have introduced a technique for applying fractal transforms in combination with other image coding methods. It could be used to improve any image coding system in which the original and approximated images are available to the coder, including wavelet compression. Once an image has been coded by the basis approximation, one can examine any partition of the image to decide where a fractal term will improve the rate/distortion characteristic.

We have incorporated a fractal term in the test examples only where the L2 measured rate/distortion performance is not degraded according to our prediction. While the PSNR is slightly improved over the whole image, in the blocks actually coded with a fractal the improvement will be several times larger, and the visual quality contributed by greater edge definition can be striking, as the examples show. This suggests that the condition for inclusion of a fractal term might be too severe, if the objective is optimum visual quality. It might be useful, for example, to select fractal enhancement for blocks with significant edges even if the PSNR is slightly degraded.

The implicit fractal method is particularly powerful because the offset information can be determined by the decoder from its reconstruction of the basis approximation. Over an entire picture, if only a proportion of blocks are fractally enhanced at an average compression penalty of 2 bits per block, the overall cost per pixel will be negligible.

bpp	PSNR Basis Approx	PSNR Centred Child	PSNR Implicit fractal	Fractal % of Picture Area	Fractal terms bpp
0.6	31.39	31.50	31.53	16.1	.028
0.5	30.74	30.88	30.90	16.1	.123
0.4	30.01	30.16	30.18	15.4	.018
0.3	29.15	29.33	29.35	14.8	.014
0.2	28.09	28.28	28.30	13.0	.009
0.1	26.60	26.70	26.71	8.1	.004

Table 1. implicit fractal applied to the standard test image Gold Hill.

5. ACKNOWLEDGEMENTS

P.Wakefield is supported by a University of Bath Studentship, with additional support from DigiMedia Vision (DMV) Ltd. D. M. Bethel is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) with additional support from the UK Independent Television Commission (ITC).

6. REFERENCES

[1] Barnsley, M. F., Ervin, V., Hardin, D., and Lancaster, J., 'Solution of an inverse problem for fractals and other sets', Proc. Nat. Acad. Sci., vol. 83, pp. 1975-1977, 1986.

[2] Jacquin, A. E., 'A novel fractal block-coding technique for digital images', Proc. ICASSP 1990, pp. 2225-2228.

[3] A. E. Jacquin, 'Fractal coding: a review', Proc. IEEE, Vol. 81, No. 10, October 1993, pp. 1451-1465.

[4] Saupe, D. and Hamzaoui, R., 'A review of the fractal image compression literature', Computer Graphics Vol. 28, No. 4, 1994, pp 268-279.

[5] Monro. D. M., and Dudbridge, F., 'Rendering Algorithms for Deterministic Fractals', IEEE Computer Graphics, Vol. 15, No. 1, January 1995, pp. 32-41.

[6] Saupe, D., 'Accelerating fractal image compression by multi-dimensional nearest neighbour search', Proceedings IEEE Data Compresion Conference (DCC95), March 1995.

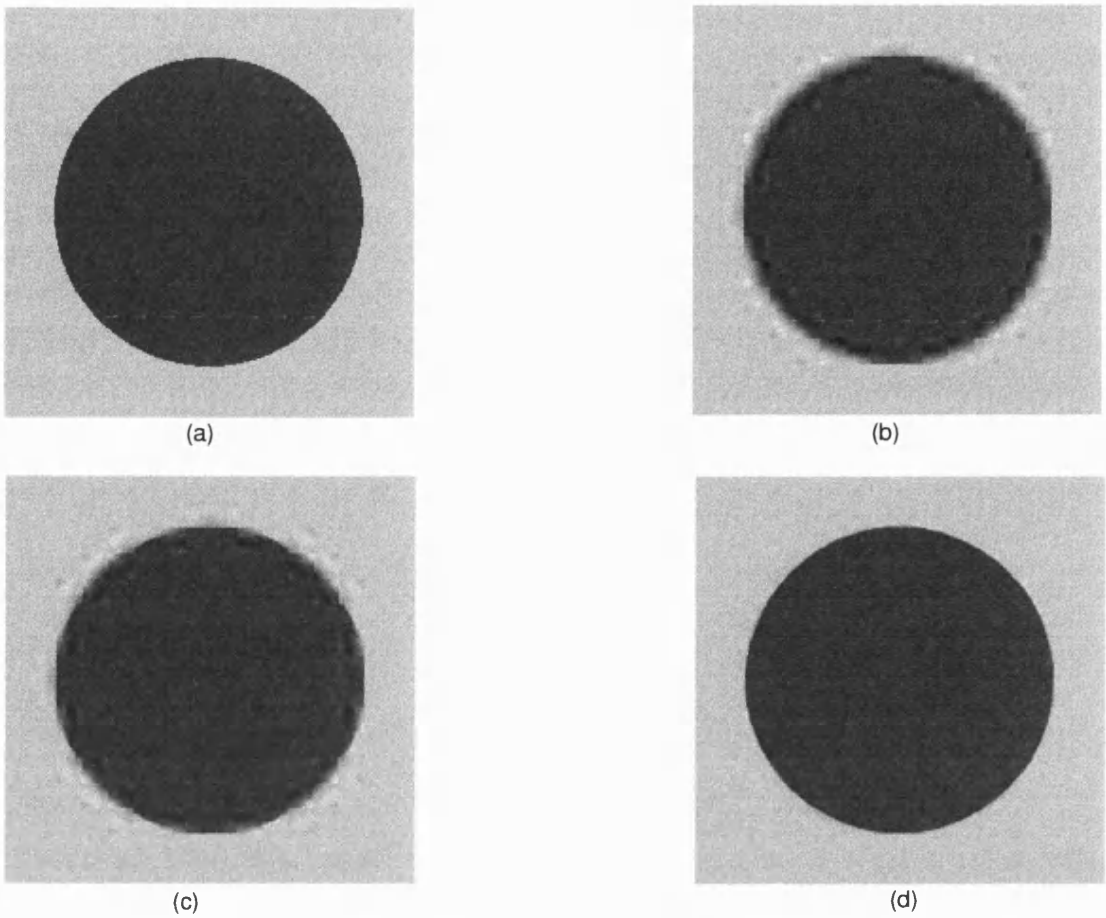
[7] Øien, G. E., Lepsøy, S. and Ramstad, T., 'Reducing the complexity of a fractal-based image coder', Proc. Eur. Signal Proc. Conf. (EUSIPCO), pp. 1353-1356, 1992.

[8] Monro, D. M., And Dudbridge, F. : 'Fractal approximation of image blocks', Proc. ICASSP 1992, pp. III: 485-488

[9] Monro, D. M., 'A hybrid fractal transform', Proc. ICASSP 1993, pp. V:169-172.

[10] Monro, D. M. and Woolley, S. J., 'Optimum parameters for hybrid fractal image coding', Proc. ICASSP 1995, 4, pp. 2571-2574.

[11] Monro, D. M. and Nicholls, J. A., 'Low bit rate colour fractal video', Proc. ICIP 95, p.III: 264-267.



(a) (b) (c) (d)

Figure 5. Edge enhancement with implicit fractals, fixed 16x16 blocks. (a) Original. (b) Basis approximation. (c) Local fractal with child centred on parent. (d) Local fractal with implicit alignment.

(a) DCT basis, PSNR 29.14 (b) Implicit fractal, PSNR 29.21 (c) Blocks selected for Implicit Fractal

Figure 6. Detail at 0.2 bpp. The PSNR difference is small, but the implicit fractal improves edges visually.

Zooming with Implicit Fractals

D. M. Monro and P. D. Wakefield

School of Electronic and Electrical Engineering, University of Bath
Claverton Down, BA2 7AY, England
e-mail: D.M.Monro@bath.ac.uk paulw@ee.bath.ac.uk
Fax: +44 1225 826073
Internet: http://dmsun4.bath.ac.uk/

ABSTRACT

One advantage of fractal image compression schemes is their multiresolution properties. An image can be decoded at higher or lower resolutions than the original, and it is possible to 'zoom-in' on sections of the image. Even so the problem of fractal zooming has received very little attention. In this paper we examine these multi-resolution properties. We study the problem of fractal zooming in general and in particular with a hybrid fractal transform with implicit fractal terms. When decoding at a resolution higher than the original, artefact can be created which affects the visual quality of the zoomed image. We present one solution to the problem of obtaining clear and sharp edges where the original approximation quality was good. Our enhanced implicit fractal transform method compares favourably with fractal zooming using previous standard fractal approximation algorithms.

1. BACKGROUND

Much interest has focused on the Iterated Function System (IFS) as a method of image coding, the theory of which is widely available in the literature. [1-6]. To compress an image, define an IFS to be $W = \{w_k; k = 1,..., N\}$, where the w_k are contraction mappings, each defined on a subset A_k of the image support S.

The attractor of W is a non-overlapping tiling of the image. A fractal function $f(x, y)$, is then defined which approximates the image brightness $g(x, y)$. An image block taken from the location A_k is referred to as the parent and its mapping $w_k(A_k)$ is referred to as the child. For each tile the brightness function is specified by a recursive mapping v_k such that

$$f(w_k(x, y)) = v_k(x, y, f(x, y)) \text{ for } (x, y) \text{ in } A_k. \quad (1)$$

In this work we use mappings of the form

$$v_k(x_0 + \delta x, y_0 + \delta y, f) = p_k(x_0 + \delta x, y_0 + \delta y) + e_k \tilde{f}_k(\delta x, \delta y) \quad (2)$$

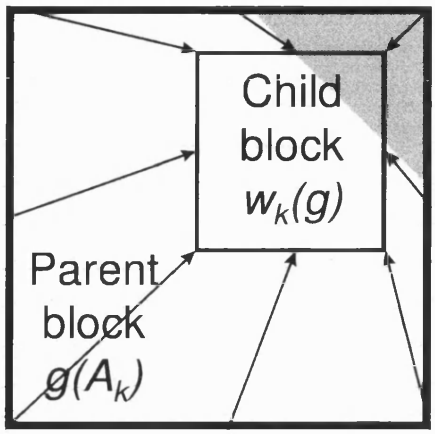


Figure 1. A local fractal transform, with an edge aligned in the parent and child blocks.

where (x_0, y_0) is the bottom left corner of A_k and $p_k(x_0 + \delta x, y_0 + \delta y) = \sum_{i=1}^{n} c_i b_i(\delta x, \delta y)$, is an approximation by basis functions $\{b_i\}$, e_k is the single fractal coefficient and $\tilde{f}_k$ is the parent block $f_k(\delta x, \delta y) = f(x_0 + \delta x, y_0 + \delta y)$ orthogonalized with respect to the basis. To solve, the known image g is used in place of the unknown fractal function f and the approximation is known to be valid by the Collage Theorem if suitable conditions are satisfied [1].

Usually the tiling of the image is by square or rectangular child blocks, and it is often assumed that p_k is a simple brightness level. Much work has concentrated on reducing the complexity of searching for the best parent to map onto each child [6, 7]. An alternative approach uses more complex basis functions [7, 8] and can restrict or even eliminate searching. With the Bath Fractal Transform (BFT) [9, 10] a pre-determined tiling without searching gives the greatest accuracy at a given compression ratio, when used with a quadratic basis. In combination with the Accurate Fractal Rendering Algorithm (AFRA) [5], the BFT has been used for real time fractal video [11].

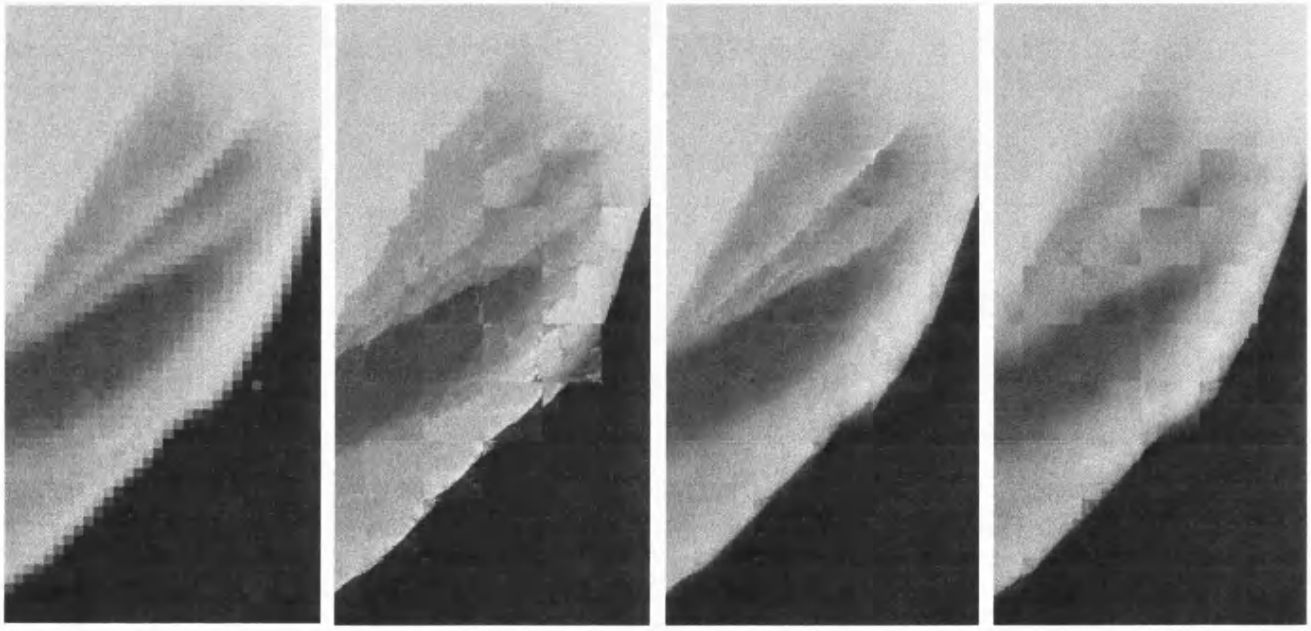
Figure 2. Lena image with zoomed areas marked.

A recent development [12] is the use of an implicit parent block location which the decoder can determine from the basis coefficients. If the existence of an edge is assumed, its location can be calculated from the ratios of the low order DCT coefficients using a pre-generated look-up table. The parent block location is then computed so that the edge is aligned in the shrunken parent and child, as in Figure 1. This implicit parent method has been shown to give a striking improvement in visual fidelity on some images. In [12] the implicit fractal was used in a near-optimal implementation in the rate-distortion sense. Table 1 shows measurements taken on the Lena image. The PSNR is slightly higher than that obtained by Fisher and Menlove at similar compression ratios [13], with much lower coder complexity (at 0.2 bpp, 2.8 sec. on a 200 MHz Pentium compared with 1122.1 sec on a Silicon Graphics IRIS 4D/35.)

2. FRACTAL ZOOMING

The fractal encoding is independent of resolution and a finite resolution decoding is performed with iterates produced at the desired resolution. In this manner the image or part of the image may be decoded to a higher resolution than the original, producing a 'zoomed' image.

To determine the effects of zooming we chose several fractal transforms that would result in a high fidelity approximation to the original. We used fixed size 6x6 child



(a) (b) (c) (d)

Figure 3. Zooming x16 (x256 pixels) in fractally encoded images with fixed 6x6 child blocks. (a) Original image. enlarged by pixel replication. (b) Zoomed fractal with DC basis and local searching.(c) Zoomed fractal with limited cosine basis and local searching. (d) Zoomed implicit fractal.

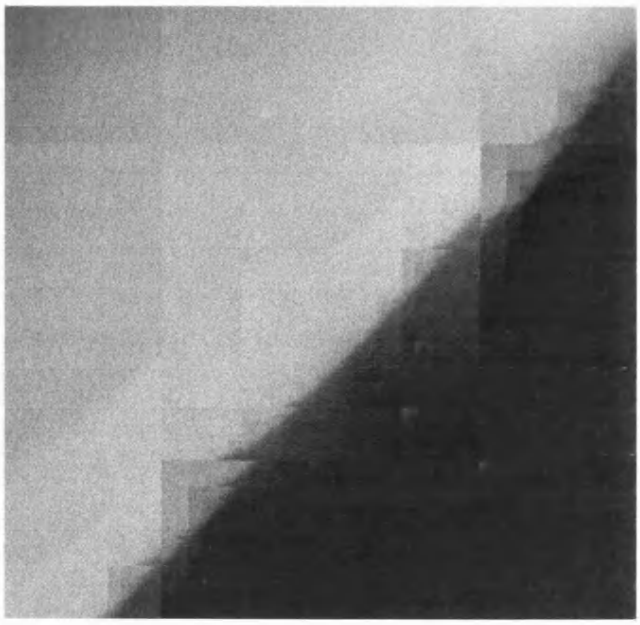
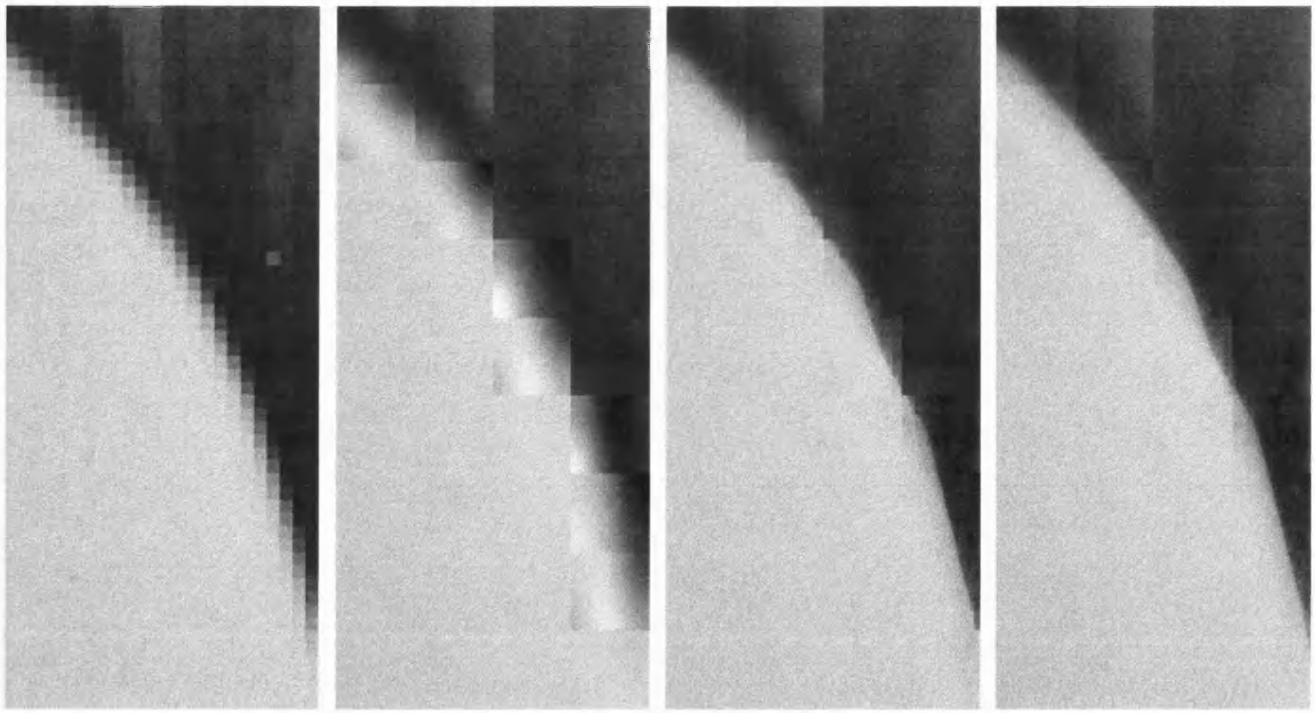


Figure 4. Cose up of edge in Figure 3(c), showing misalignment at block boundaries.

bpp	PSNR Basis Approx	PSNR Centred Child	PSNR Implicit fractal	Fractal % of Area	Fractal terms bpp
0.6	35.35	35.29	35.57	10.5	.016
0.5	34.62	34.53	34.80	15.4	.017
0.4	33.68	33.68	33.83	10.1	.010
0.3	32.43	32.42	32.66	13.2	.009
0.2	30.71	30.70	31.07	16.9	.007
0.1	28.13	28.09	28.35	26.3	.005

Table 1. Implicit fractal applied to Lena.

blocks with searching over ±5 child block widths for the best 12x12 parent. We implemented a standard fractal transform as invented by Jacquin [2] with a simple DC basis, and also a version with a 2D DCT basis limited to the 6 lowest order terms. We compared these with an implicit fractal approximation [12] also using a fixed 6x6 block size partition. Because the distance of the parent from the child is limited in extent, when zooming we need to render only the area of interest extended by 5 child blocks, rather than the



(a) (b) (c) (d)

Figure 5. Fractal zooming x16 and enhanced fractal zooming with implicit fractals, using fixed 6x6 child blocks. (a) Original image with pixel replication. (b) Zoomed basis image. (c) Implicit fractal zoom. (d) Enhanced implicit fractal zoom.

entire image which would be huge. This greatly reduces the computation cost. Figure 2 marks two areas of interest. The results of zooming on the upper one of these by a factor of 16 in each spatial dimenstion (giving 256 times as many pixels) for each transform are displayed in Figure 3.

From these results we make several observations:

1. When edges are well approximated at the original resolution they are sharp and fairly well preserved in the zoomed images

2. The 'traditional' DC basis zoom produces significant distortion, while the more complex basis, as used in the BFT, gives more stable decoding for both the local searching transform and the implicit fractal

3. Edges do not always match well at block boundaries (see figure 4).

3. ENHANCED ZOOMING WITH IMPLICIT FRACTALS

In [12] the implicit fractal transform, used in a optimal quad-tree implementation provided excellent rate/distortion results. It was also shown that the performance of fractal coding schemes is improved by using implicit parent locations because they have zero bit-cost. As a result, we give the implicit fractal transform special consideration for zooming.

In [12] fractal terms were only used in a minority of blocks, in which they improved the PSNR of the compressed image, as in Table 1. When zooming, the non-fractal blocks are less visually satisfactory. For example, in Figure 5 (c), one block near the top left corner does not have a fractal term despite being part of an edge. Elsewhere along Lena's shoulder, 'notches' are created by non-fractal blocks, which are propagated by iteration onto neighbouring blocks.

We improve the implicit fractal zoom by introducing additional fractal terms at the decoding stage. Using the mathematical edge model from the coding stage, we estimate where edges in fractal blocks will intersect non-fractal blocks. If two edges meet the sides of a non-fractal block, we assume that block had an edge in the original which was judged to be too insignificant in the MSE sense to be coded, or which was lost when the image was digitised. We then compute a parent location and fractal coefficient based on the location of the intersecting edges and the coefficients of the fractal blocks. The improvement may be seen in Figure 5 (d).

4. DISCUSSION AND CONCLUSIONS

We have zoomed on images encoded with several different fractal techniques commonly employed for image coding and compression. The zooming process worked as expected and all the fractal zooms we produced looked better than pixel replication from the original resolution. The fractal zoom provides sharp edges in each of the edge blocks which are reasonably well preserved in the normal decoding, although edges between blocks do not always line up. Increasing the number of fractal blocks improved the visual quality of the zoomed implicit fractal transform.

Detail introduced by zooming is not present in the original pictures, so there is no 'original' with which to compare. As a result evaluation must be subjective, based on whether the new detail is visually an acceptable extension of the original. Assuming the original approximation quality is good we conclude that the zoomed image can be visually superior to pixel replication. We believe zooming on edges is a key problem in the development of fractal zooming. We see this as an important area of interest in further work.

5. ACKNOWLEDGEMENTS

P. D. Wakefield is supported by a University of Bath Studentship, with additional support from Network Digital Systems (NDS) Ltd. The authors would like to thank David Bethel for providing rate/distortion results and Gary Dickson for preparation of the drawings.

6. REFERENCES

[1] Barnsley, M. F., 'A better way to compress images', BYTE, Vol. 13, No. 1, pp. 215-224, January 1988.

[2] Jacquin, A. E., 'A novel fractal block-coding technique for digital images', Proc. ICASSP 1990, pp. 2225-2228.

[3] Jacquin, A. E., 'Fractal coding: a review', Proc. IEEE, Vol. 81, No. 10, October 1993, pp. 1451-1465.

[4] Saupe, D. and Hamzaoui, R., 'A review of the fractal image compression literature', Computer Graphics Vol. 28, No. 4, 1994, pp. 268-279.

[5] Monro, D. M., and Dudbridge, F., 'Rendering algorithms for determinisitic fractals', IEEE Computer Graphics, Vol. 15, No. 1, January 1995, pp. 32-41.

[6] Monro, D. M., and Dudbridge, F., 'Fractal approximation of image blocks', Proc. ICASSP 1992, pp. III: 485-488.

[7] Øien, G. E., Lepsøy, S. and Ramstad, T., 'Reducing the complexity of a fractal-based image coder', Proc. Eur. Signal Proc. Conf. (EUSIPCO), pp. 1353-1356, 1992

[8] Saupe, D., 'Accelerating fractal image compression by multi-dimensional nearest neightbour search', Proceedings IEEE Data Compression Conference (DCC95), March 1995.

[9] Monro, D. M., 'A hybrid fractal transform', Proc. ICASSP 1993, pp. V:169-172.

[10] Monro, D. M., and Woolley, S. J., 'Optimum parameters for hybrid fractal image coding', Proc. ICASSP 1995, 4, pp. 2571-2574.

[11] Monro, D. M., and Nicholls, J. A., 'Low bit-rate colour fractal video', Proc. ICIP 95, p.III: 264-267.

[12] Wakefield, P. D., Bethel, D. M., and Monro, D. M., 'Hybrid image compression with implicit fractal terms', Proc. ICASSP 1997, pp. IV-2933-2936.

[13] Fisher, Y. and Menlove, S., 'Fractal encoding with HV partitions', in Fractal Image Compression, Fisher, Y. ed., Springer-Verlag, 1995.

Fractal Enhancement of Decompressed Images

P. D. Wakefield and D. M. Monro
Department of Electronic and Electrical Engineering, University of Bath
Claverton Down, BA2 7AY, UK. Fax: +44 1225 826073
e-mail: paulw@ee.bath.ac.uk d.m.monro@bath.ac.uk
Internet: http://dmsun4.bath.ac.uk

Abstract

Fractal image compression successfully uses self-similarity within an image to achieve compression. By adapting fractal image compression techniques to image enhancement, an innovative new fractal technique for visual enhancement of JPEG compressed images is created. After detecting the self-similarity characteristics of a degraded JPEG image, details spoiled by quantization are enhanced using an iterative scheme and the fractal nature of the method allows resolution enhancement and fractal zooming. The effect of the scheme is an increase in the visual quality of edges in the image for almost all areas and sharp results when the image resolution is increased.

This application of fractal theory to the problem of image enhancement reveals a number of interesting avenues of investigation and promises techniques which can be used in combination with conventional methods, exploiting both self-similarity and statistical properties to achieve enhancement.

1. Background

Compression of still images with the JPEG standard [1] is important for many applications which depend on media with limited capacity. At compression ratios of up to 10 to one and sometimes higher only a small perceptual deterioration in image quality occurs. At greater compression ratios however the perceptual quality decreases rapidly due to the greater quantization of DCT frequency coefficients. Quantization causes loss of image details and blocking and consequently several techniques have been proposed to reduce these aretfacts. Primarily these have been aimed at reducing blockiness whilst attempting to retain detail [2,3].

Fractal methods provide a new potential source of image enhancement techniques. Images can be compressed using fractals by storing information about the self-similarity of an image, resulting in a fractal code which is independent of the resolution of the source image and is efficient enough to allow high compression ratios. We show here that the methods of fractal image compression may be adapted so that the self-similarity of an image can be used for image enhancement. Because of the mutli-resolution properties of fractals this method can also be used to increase the image resolution. We investigate the results for various compression ratios and find that fractal transforms, originally developed for compression, are useful for enhancement of JPEG compressed images.

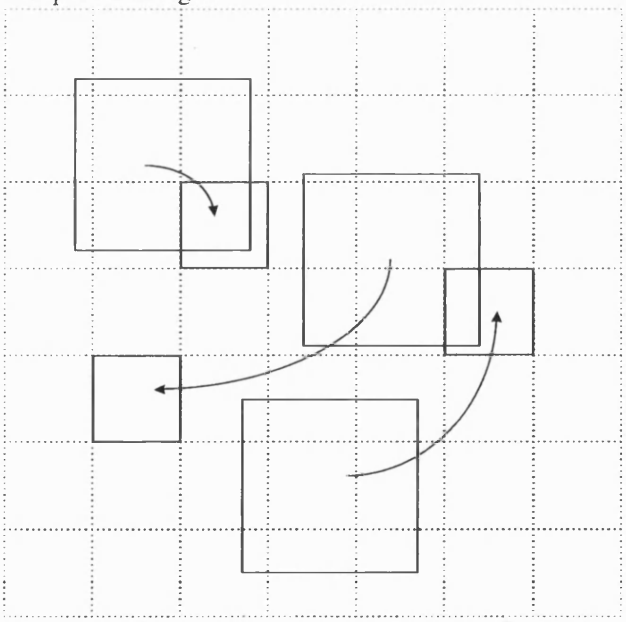


Figure 1. Some example parent and child blocks, whose relationship describes the self-similarity of the image.

2. Method

The method introduced here has two stages. Firstly the self-similarity characteristics of the JPEG image are

determined using a procedure based on the encoding stage of a local search fractal transform [5-6]. Secondly an iterative procedure is applied to the image to restore frequency information lost during the quantization process.

2.1 Characterisation Stage

To characterise the image a square partition of the image is created with 8x8 pixel child blocks. For each child block C a 16x16 pixel parent block location is determined from all blocks of that size within a certain pixel distance of the child as illustrated by Figure 1.

To determine the location each possible parent is taken in turn, is decimated to the size of the child and has its DC component removed. The parent is then scaled to match the child as closely as possible. I.e.

$$\|C-\alpha P\| \qquad (1)$$

is minimised. The parent location which gives the largest effect is chosen, so that $|\alpha|\|P\|$ is maximal. But alternatively the parent which best minimises (1) can be chosen. The result is a parent location and fractal coefficient α for each child, representing the piecewise self-similarity of the image.

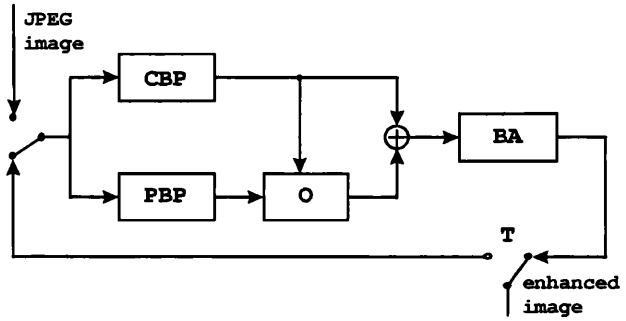


Figure 2: Operation of enhancement function. CBP - child block provider; PBP - parent block provider; O - orthogonalisation operator; BA - block assembler; T - threshold switch.

2.2 Enhancement Stage

Using the self-similarity information determined above, a function τ is defined which operates on the space of images.

For an image x, $\tau(x)$ is constructed by copying the original JPEG image s then adding fractal components. For each cell in the partition a child block C is taken from s at that location and a parent P is taken from x at the corresponding parent location. This block is contracted to the size of the child by averaging and then has its DC component and child block component removed by Gram-Schmitt orthogonalisation [7]:

$$\tilde{P} = P - \langle P,\mathbf{1}\rangle\mathbf{1} - \langle P,\tilde{C}\rangle\tilde{C} \qquad (2)$$

where $\mathbf{1}$ is the matrix of 1's and $\tilde{C} = C - \langle C,\mathbf{1}\rangle\mathbf{1}$. $\langle X,Y\rangle$ denotes the inner product of image blocks.

The block $\alpha\tilde{P}$ is then the fractal component added, where α is the fractal coefficient corresponding to that child block. It should be noted that it is required that τ be a contraction mapping so that the iteration procedure described below will be convergent. Guaranteeing this property is a problem which remains largely unsolved in fractal image compression, but in practice this is not usually a cause for concern. From our experience it is reasonable to require $|\alpha| < 5$.

Enhancement of the image requires a finite sequence of iterates to be computed starting with the original image s and using the function τ:

$$s, \tau(s), \tau^2(s), \ldots, \tau^M(s) \qquad (3)$$

The number of iterations, M, is chosen large enough for the sequence to effectively converge and for this purpose we use $M = \lfloor \log_2 m \rfloor$, where m is the largest child dimension. If the enhanced image is to be the same dimensions as the original then $m = 8$ and the number of iterates is 3. If decoding at higher resolutions [6] the underlying JPEG image must now be resized and the determined parent locations scaled appropriately. The number of iterations is then calculated using the above formula. The M^{th} iterate $\tau^M(s)$ is the final, enhanced version of the image.

The operation of τ is illustrated in Figure 2.

3. Evaluation

To evaluate the method we enhanced the grey-scale test images Lena, Gold Hill, Barbara 2, and Boats after compression at 10:1, 20:1, 30:1, and 40:1. We allowed a distance of 8 pixels between the centres of the parent and child and in (1) we used the L2 norm with corresponding inner product in (2).

The result was a noticeable improvement in the blurred edges and a slight reduction in blockiness in flat areas. A section of the Gold Hill result at 20:1 is shown in Figure 3. Only rarely did the fractal method reduce visual quality.

To appraise the multi-resolution properties of the method we rendered the enhanced 20:1 Gold Hill image at x2 resolution. This result is also shown in Figure 3. In this image edges are sharper and there is no extra

pixelation. Finer texture has been introduced by the self-similarity mapping.

To demonstrate the compatibility of this method with algorithms for reducing blockiness we applied a very simple averaging filter at the block edges after the fractal enhancement procedure was completed. The result was a considerable reduction in blockiness and retention of sharpened edges. Figure 4 shows the results for a section of the Lena image.

To enhance an image takes about two minutes. To produce a faster method we can replace the searching part of the algorithm with the implicit fractal method presented in [4]. This results in less visual improvement than the basic method, but the algorithm is considerably faster.

4. Conclusions

We have demonstrated a method for enhancement of JPEG images which uses a fractal technique adapted from fractal image compression. The method requires no information from the original image. The method dramatically increases sharpness of edges and details. Although a small number of blocks are made worse by the procedure and the PSNR of the enhanced images compared with the uncompressed source is lower, the overall effect is an increase in the visual fidelity of JPEG compressed images in all the examples tried. In resolution enhancement the visual quality is improved and it is possible to 'zoom-in' on particular areas of an image (see [6]). We have demonstrated this method in combination with a de-blocking technique and see no reason why more advanced methods should not also be used. The fractal method providing a natural complement to conventional methods.

The primary application of this method is in improving the visual quality of compressed images, either on its own or in combination with other methods. We have demonstrated it here on images degraded by JPEG coding, but it may be adapted to work on images of various kinds. There exist many compressed images where the original is no longer available. This technique should work whenever a degradation process leaves self-similarity intact. At this stage of development this technique shows promise for use in a number of areas.

5. Acknowledgements

P. D. Wakefield is supported by a University of Bath Studentship, with additional support from News Digital Systems (NDS) Ltd.

6. References

[1] Wallace, G. K., "The JPEG still picture compression standard", *Comm. ACM*, 1991, 34, (4), pp. 30-40

[2] Horng, R. and Ahumada, A. J., Jr., "A fast DCT block smoothing algorithm", *Visual Communication and Image Processing Annual Meeting Proceedings*, SPIE Vol. 2501, Paper 5, 1995

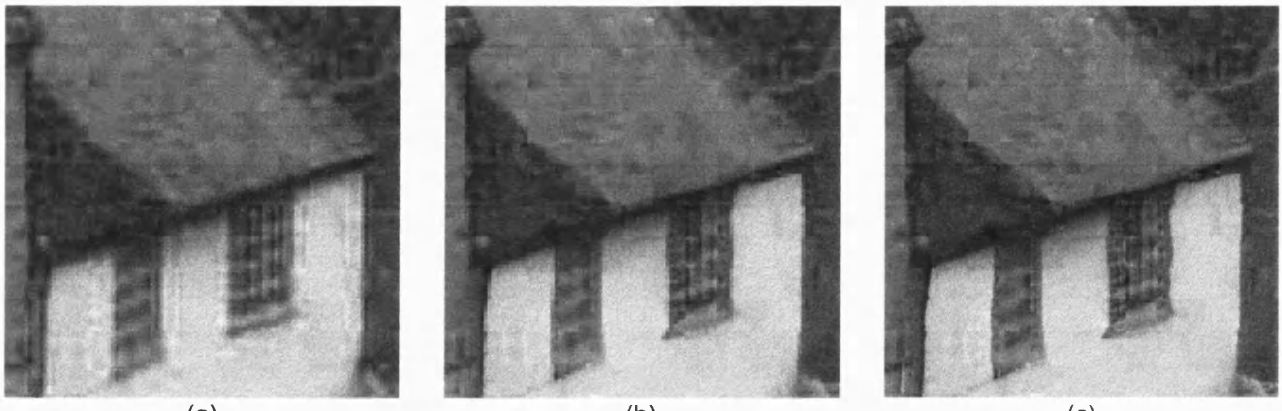
[3] Gopinath, R. A., Lang, M., Guo, J. and Odegard, J. E., 'Enhancement of decompressed images at low bit rates', *Proc. SPIE, Mathematical Imaging: Wavelet Applications in Signal and Image Processing II (2303-30)*, July 27-29, 1994, San Diego, CA.

[4] Wakefield, P. D., Bethel, D. M. and Monro, D. M., 'Hybrid image compression with implicit fractal terms', *Proc. ICASSP 97*, pp. IV-2933-2936.

[5] Jacquin, A. E., 'Fractal coding: a review', *Proc. IEEE*, Vol. 81, No. 10, October 1993, pp. 1451-1465.

[6] Monro, D. M. and Wakefield, P. D., 'Zooming with implicit fractals', *Proc. ICIP 97*, pp. I-913-916.

[7] Kreyszig, E., *Introduction to functional analysis with*



(a) (b) (c)

Figure 3. Fractal enhancement of JPEG Gold Hill image. (a) JPEG image at 20:1 compression. (b) Image after fractal enhancement. (c) Image after fractal enhancement and fractal resolution enhancement.

applications, Robert E. Krieger Publishing Company, Malabar, FL, 1989

(a) (b)

Figure 4. Fractal enhancement with smoothing of JPEG Lena image. (a) JPEG image at 20:1 compression. (b) Image after fractal enhancement and smoothing.

Appendix B

DERIVATION OF IMPLICIT PARENT LOCATIONS FOR A POLYNOMIAL BASIS

Edge Model

In order to determine the optimal parent location a simple edge model is used:

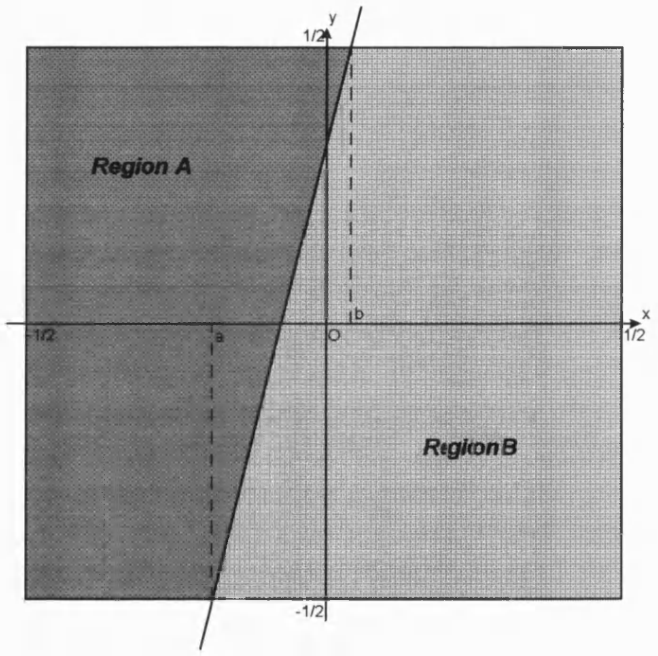


Figure B.1. Edge model used for derivation of parent location.

Given a basis approximation to a child block the parameters a and b are determined using the model above. This gives the location of a straight line through the child block and the parent block is placed so that the mid-point of this line is a common point in both the parent and child.

Mid-point

The midpoint is denoted $\underline{m}$. If $b > a \geq -0.5$

$$\underline{m} = \left(\frac{b+a}{2}, 0\right)$$

If $b > -0.5 > a$

$$\underline{m} = \left(\frac{2b-1}{4}, \frac{-2a-1}{4(b-a)}\right)$$

Parent location

The parent location is

$$\underline{p} = \underline{c} - k\left(\underline{m} + \left(\frac{1}{2}, \frac{1}{2}\right)\right)$$

where k is the block size and $\underline{c}$ is the location of the bottom left corner of the child.

Basis Approximation

The basis approximation has the form

$$I(x,y) \approx c_0 + c_1 2\sqrt{3}x + c_2 2\sqrt{3}y + c_3 6\sqrt{5}\left(x^2 - 1/12\right) + c_4 12xy + c_5 6\sqrt{5}\left(y^2 - 1/12\right)$$

where $I(x,y)$ is the image intensity. Each of the coefficients is calculated by inner product of its corresponding basis function with the image block $I(x,y)$. For example

$$c_1 = \int_{-0.5}^{0.5}\int_{-0.5}^{0.5} I(x,y) \cdot 2\sqrt{3}x\,dy\,dx\,.$$

Derivation for c_1

For $b > a \geq -0.5$ the integral over region A is

$$\int_{x=-0.5}^{a} 2\sqrt{3}x \int_{y=-0.5}^{0.5} dydx + \int_{x=a}^{b} 2\sqrt{3}x \int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} dydx$$

$$= 2\sqrt{3}\left(\int_{-0.5}^{a} xdx + \int_{x=a}^{b} \frac{x}{2} + \frac{-2x^2 + ax + bx}{2(b-a)} dx \right)$$

$$= \frac{2\sqrt{3}\left(4\left(b^3 - a^3\right) - 3(b-a)\right)}{24(b-a)}$$

Because the basis function is orthogonal with respect to the DC function, the integral over one area is equal to the negative of the integral over the complement when I is constant. As a result

$$c_1 = (l-r)\left(\frac{2\sqrt{3}\left(4\left(b^3 - a^3\right) - 3(b-a)\right)}{24(b-a)} \right)$$

A simpler explanation is to say that adding or subtracting a constant value to the block will not effect the coefficient value. Hence adding $-r$ gives $l-r$ in region A and 0 in region B, from which the above result should be clear.

For $b > -0.5 > a$ the integral over region A is

$$\int_{x=0}^{b} 2\sqrt{3}x \int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} dydx$$

$$= 2\sqrt{3}\left(\int_{x=-0.5}^{b} \frac{x}{2} + \frac{-2x^2 + ax + bx}{2(b-a)} dx \right)$$

$$= \frac{2\sqrt{3}\left(4b^3 - 1 - 3b\right)}{24(b-a)}$$

Hence

$$c_1 = (l-r)\left(\frac{2\sqrt{3}\left(4b^3-1-3b\right)}{24(b-a)}\right)$$

For $a = b$ the integral over region A is

$$\int_{x=-0.5}^{a} 2\sqrt{3}x \int_{y=-0.5}^{0.5} dydx$$
$$= 2\sqrt{3}\int_{-0.5}^{a} xdxdx$$
$$= 2\sqrt{3}\left(\frac{a^2}{2}-\frac{1}{8}\right)$$

Hence

$$c_1 = (l-r)2\sqrt{3}\left(\frac{a^2}{2}-\frac{1}{8}\right)$$

Derivation for c_2

For $b > a \geq -0.5$ the integral over region A is

$$\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5} 2\sqrt{3}ydydx + \int_{x=a}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} 2\sqrt{3}ydydx$$
$$= 2\sqrt{3}\int_{x=a}^{b} \frac{1}{8} + \frac{-4x^2+4ax+4bx-a^2-2ab-b^2}{8(b-a)^2}dx$$
$$= \frac{2\sqrt{3}(b-a)^2}{12(b-a)}$$

Hence

$$c_2 = (l-r)\left(\frac{2\sqrt{3}(b-a)^2}{12(b-a)}\right)$$

For $b > -0.5 > a$ the integral over region A is

$$\int_{x=-0.5}^{b}\int_{y=\frac{x-a-b}{2(b-a)}}^{0.5} 2\sqrt{3}ydydx$$
$$= 2\sqrt{3}\int_{x=-0.5}^{b}\frac{1}{8}+\frac{-x^2+2ax+2bx-a^2-2ab-b^2}{8(b-a)^2}dx$$
$$= \frac{2\sqrt{3}\left(4b^3-12ab^2-12ab-3a-3b-1\right)}{48(b-a)^2}$$

Hence

$$c_2 = (l-r)\left(\frac{2\sqrt{3}\left(4b^3-12ab^2-12ab-3a-3b-1\right)}{48(b-a)^2}\right)$$

For $a = b$ the integral over region A is

$$\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5} 2\sqrt{3}ydydx$$
$$= 0$$

Hence

$$c_2 = 0\,.$$

Derivation for c_3

For $b > a \geq -0.5$ the integral over region A is

$$\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5} 6\sqrt{5}\left(x^2-1/12\right)dydx+\int_{x=a}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} 6\sqrt{5}\left(x^2-1/12\right)dydx$$

$$=-6\sqrt{5}\frac{b+a+1}{24}+6\sqrt{5}\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5}x^2dydx+6\sqrt{5}\int_{x=a}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5}x^2dydx$$

$$=-6\sqrt{5}\frac{b+a+1}{24}+6\sqrt{5}\int_{-0.5}^{a}x^2dx+6\sqrt{5}\int_{x=a}^{b}\frac{x^2}{2}+\frac{-2x^3+ax^2+bx^2}{2(b-a)}dx$$

$$=-6\sqrt{5}\frac{b+a+1}{24}+6\sqrt{5}\frac{2\left(b^4-a^4\right)+b-a}{24(b-a)}$$

$$=\frac{6\sqrt{5}\left(2\left(b^4-a^4\right)+a^2-b^2\right)}{24(b-a)}$$

Hence

$$c_3=(l-r)\left(\frac{6\sqrt{5}\left(2\left(b^4-a^4\right)+a^2-b^2\right)}{24(b-a)}\right)$$

For $b>-0.5>a$ the integral over region A is

$$\int_{x=-0.5}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} 6\sqrt{5}\left(x^2-1/12\right)dydx$$

$$=6\sqrt{5}\int_{x=-0.5}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5}x^2dydx-6\sqrt{5}\frac{(2b+1)^2}{96(b-a)}$$

$$=6\sqrt{5}\int_{x=-0.5}^{b}\frac{x^2}{2}+\frac{-2x^3+ax^2+bx^2}{2(b-a)}dx-6\sqrt{5}\frac{(2b+1)^2}{96(b-a)}$$

$$=6\sqrt{5}\left(\frac{16b^4+8b+3}{192(b-a)}-\frac{(2b+1)^2}{96(b-a)}\right)$$

$$=\frac{6\sqrt{5}\left(4b^2-1\right)^2}{192(b-a)}$$

Hence

$$c_3 = (l-r)\left(\frac{6\sqrt{5}(4b^2-1)^2}{192(b-a)}\right)$$

For $a = b$ the integral over region A is

$$\begin{aligned}
&\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5} 6\sqrt{5}\left(x^2 - 1/12\right)dydx \\
&= 6\sqrt{5}\int_{x=-0.5}^{a} x^2 - 1/12\,dx \\
&= 6\sqrt{5}\left(\frac{a^3}{3} - \frac{a}{12}\right)
\end{aligned}$$

Hence

$$c_3 = (l-r)6\sqrt{5}\left(\frac{a^3}{3} - \frac{a}{12}\right).$$

Derivation for c_4

For $b > a \geq -0.5$ the integral over region A is

$$\begin{aligned}
&\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5} 6\sqrt{5}\left(y^2 - 1/12\right)dydx + \int_{x=a}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} 6\sqrt{5}\left(y^2 - 1/12\right)dydx \\
&= -6\sqrt{5}\frac{b+a+1}{24} + 6\sqrt{5}\int_{x=-0.5}^{a}\int_{y=-0.5}^{0.5} y^2 dydx + 6\sqrt{5}\int_{x=a}^{b}\int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} y^2 dydx \\
&= -6\sqrt{5}\frac{b+a+1}{24} + 6\sqrt{5}\int_{x=-0.5}^{a} 1/12\,dx \\
&+ 6\sqrt{5}\int_{x=a}^{b} \frac{-b^3 - 3a^2b + 4x^3 - 6ax^2 - 6bx^2 + 3a^2x + 6abx + 3b^2x}{12(b-a)^3}dx \\
&= -6\sqrt{5}\frac{b+a+1}{24} + 6\sqrt{5}\frac{b+a+1}{24} \\
&= 0
\end{aligned}$$

Hence

$$c_4 = 0$$

For $b > -0.5 > a$ the integral over region A is

$$\int_{x=-0.5}^{b} \int_{y=\frac{2x-a-b}{2(b-a)}}^{0.5} 6\sqrt{5}\left(y^2 - 1/12\right) dy dx$$

$$= 6\sqrt{5} \int_{x=-0.5}^{b} \int_{y=\frac{x-a-b}{2(b-a)}}^{0.5} y^2 dy dx - 6\sqrt{5}\frac{(2b+1)^2}{96(b-a)}$$

$$= 6\sqrt{5} \int_{x=-0.5}^{b} \frac{-b^3 - 3a^2b + 4x^3 - 6ax^2 - 6bx^2 + 3a^2x + 6abx + 3b^2x}{12(b-a)^3} dx - 6\sqrt{5}\frac{(2b+1)^2}{96(b-a)}$$

$$= \frac{\sqrt{5}\left(16a^2b^2 + 4b^2 + 16a^2b + 1 + 4a + 4b + 4a^2 + 16ab + 16ab^2\right)}{32(b-a)^3}$$

Hence

$$c_4 = (l-r)\left(\frac{\sqrt{5}\left(16a^2b^2 + 4b^2 + 16a^2b + 1 + 4a + 4b + 4a^2 + 16ab + 16ab^2\right)}{32(b-a)^3}\right)$$

For $a = b$ the integral over region A is

$$\int_{x=-0.5}^{a} \int_{y=-0.5}^{0.5} y^2 - 1/12 \, dy dx$$

$$= 0$$

Hence

$$c_4 = 0 .$$

Summary of Solutions

For $b > a \geq -0.5$ (confirmed with numerical results)

$$c_1 = (l-r)\left(\frac{2\sqrt{3}\left(4\left(b^3 - a^3\right) - 3(b-a)\right)}{24(b-a)}\right)$$

$$c_2 = (l-r)\left(\frac{2\sqrt{3}(b-a)^2}{12(b-a)}\right)$$

$$c_3 = (l-r)\left(\frac{6\sqrt{5}\left(2\left(b^4-a^4\right)+a^2-b^2\right)}{24(b-a)}\right)$$

$$c_4 = 0$$

For $b > -0.5 > a$ (confirmed with numerical results - except c_4)

$$c_1 = (l-r)\left(\frac{2\sqrt{3}\left(4b^3-1-3b\right)}{24(b-a)}\right)$$

$$c_2 = (l-r)\left(\frac{2\sqrt{3}\left(4b^3-12ab^2-12ab-3a-3b-1\right)}{48(b-a)^2}\right)$$

$$c_3 = (l-r)\left(\frac{6\sqrt{5}\left(4b^2-1\right)^2}{192(b-a)}\right)$$

$$c_4 = (l-r)\left(\frac{\sqrt{5}\left(16a^2b^2+4b^2+16a^2b+1+4a+4b+4a^2+16ab+16ab^2\right)}{32(b-a)^3}\right)$$

For $b = a$ (confirmed with numerical results)

$$c_1 = (l-r)2\sqrt{3}\left(\frac{a^2}{2}-\frac{1}{8}\right)$$

$$c_2 = 0$$

$$c_3 = (l-r)6\sqrt{5}\left(\frac{a^3}{3}-\frac{a}{12}\right)$$

$$c_4 = 0$$

Remember, in software these sampled functions will be normalised. This means the coefficient values will be smaller by a factor of the block size. I.e. N for an $N \times N$ block.

Look-up Table Solution

The simplest solution found to this problem uses a look-up table. First note that the intensity values l and r are not of relevance to the parent location. Only a and b need be determined. A look up table is constructed with $|c_3 / c_1|$ and $|c_2 / c_1|$ values on the left hand side and the midpoint corresponding to the a and b values on the right. The table is constructed by choosing a range of values for a and b with a uniform distribution, with a density inversely proportional to the size of the child block.

For $b > a \geq -0.5$

$$c_3 / c_1 = \frac{\sqrt{15}\left(2\left(b^4 - a^4\right) + a^2 - b^2\right)}{4\left(b^3 - a^3\right) - 3(b - a)}$$

$$c_2 / c_1 = \frac{2(b-a)^2}{4\left(b^3 - a^3\right) - 3(b - a)}$$

For $b > -0.5 > a$

$$c_3 / c_1 = \frac{\sqrt{15}\left(4b^2 - 1\right)^2}{8\left(4b^3 - 1 - 3b\right)}$$

$$c_2 / c_1 = \frac{4b^3 - 12ab^2 - 12ab - 3a - 3b - 1}{2\left(4b^3 - 1 - 3b\right)(b - a)}$$

For $b = a$

$$c_3 / c_1 = \frac{2\sqrt{15}}{3} a$$

$$c_2 / c_1 = 0 .$$

When encoding a child block it is first approximated by the basis functions so that the ratios $|c_3 / c_1|$ and $|c_2 / c_1|$ are known. The previously constructed table is now searched to find the closest matching ratios, and this gives the midpoint of the line running through the block. From this information the optimal parent location can be easily calculated (see earlier).

Use of Symmetry

The solution so far is for an edge which has an angle from the vertical of less than 45 degrees, slopes to the right, and lies to the left of the centre of the child block. Results for all other angles can be computed from the symmetry of the polynomial functions. The set of rules used to achieve this are as follows:

Firstly if $|c_1| < |c_2|$ the edge is assumed to be predominantly vertical and the algorithm is identical but the y coefficients and x coefficients are switched. Secondly by using absolute values when searching the look-up table the signs of the ratios can be subsequently used to determine a reflection of the midpoint and give a solution for other cases. If c_3/c_1 is negative and c_2/c_1 positive $\underline{m}$ is reflected in the horizontal axis. If c_3/c_1 is positive and c_2/c_1 is positive $\underline{m}$ is reflected in the vertical axis. If c_3/c_1 is positive and c_2/c_1 negative $\underline{m}$ is reflected in both axes.

Appendix C

DERIVATION OF EXTENDED IFT

Edge Model

The edge model used is shown in Figure C.1. The intensity value in regions A and C is O and in region B M.

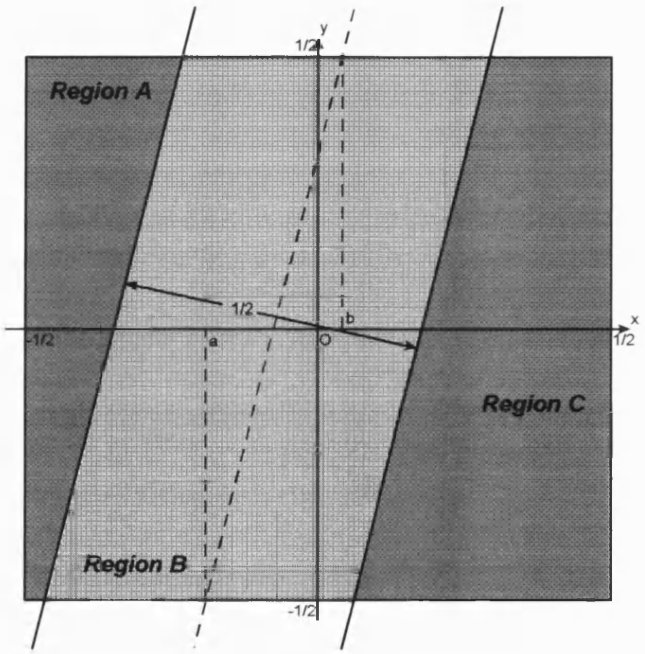


Figure C.1. Edge model used for derivation of parent location.

Intercept Points

The intercept points of the left edge with the bottom and top of the child block are denoted a_1 and b_1 respectively, for the right edge the intercepts are a_2 and b_2. For the left edge

$$a_1 = a - \frac{\sqrt{1+(b-a)^2}}{4} \text{ and } b_1 = b - \frac{\sqrt{1+(b-a)^2}}{4}$$

and for the right edge

$$a_2 = a + \frac{\sqrt{1+(b-a)^2}}{4} \text{ and } b_2 = b + \frac{\sqrt{1+(b-a)^2}}{4}.$$

Mid-point

The mid-points of the lines are given by the same formulae as in the IFT derivation.

Dual parent mapping

Two parent blocks are used - one for each edge. The common point for each parent is the midpoint of the corresponding line. Hence the location of the parent is given by the same formula as in the IFT derivation.

Each point in the child block however is only mapped onto from one parent. Which parent is used depends on which line the point in the child block is closest to.

Derivation

It may be seen that the integral is a sum of two integrals of the type derived for the IFT. In the first integral regions A and B have intensity value $M/2$ and region C intensity value $O - M/2$. In the second, region A has intensity $O - M/2$ and regions B and C have intensity $M/2$. The sum of these blocks corresponds to the edge model above, and since the sum of the integrals is equal to the integral of the sums the formulae used to derive the IFT can be used again.

Hence assuming the centre (dashed) line is within 45 degrees of the vertical, slopes to the right, and lies to the left of the origin the coefficient values will be:

Case 1: $0 \geq b = a > -0.25$,

$$c_1 = (O-M)\left(2\sqrt{3}\left(\frac{a_1^2}{2}-\frac{1}{8}\right)-2\sqrt{3}\left(\frac{a_2^2}{2}-\frac{1}{8}\right)\right)$$

$$c_2 = 0$$

$$c_3 = (O-M)\left(6\sqrt{5}\left(\frac{a_1^3}{3}-\frac{a_1}{12}\right)-6\sqrt{5}\left(\frac{a_2^3}{3}-\frac{a_2}{12}\right)\right)$$

$$c_4 = 0$$

Case 2: $0.5 \geq b_1 > a_1 \geq -0.5$,

$$c_1 = (O-M)\left(\frac{2\sqrt{3}\left(4\left(b_1^3-a_1^3\right)-3\left(b_1-a_1\right)\right)}{24\left(b_1-a_1\right)}-\frac{2\sqrt{3}\left(4\left(b_2^3-a_2^3\right)-3\left(b_2-a_2\right)\right)}{24\left(b_2-a_2\right)}\right)$$

$$c_2 = 0$$

$$c_3 = (O-M)\left(\frac{6\sqrt{5}\left(2\left(b_1^4-a_1^4\right)+a_1^2-b_1^2\right)}{24\left(b_1-a_1\right)}-\frac{6\sqrt{5}\left(2\left(b_2^4-a_2^4\right)+a_2^2-b_2^2\right)}{24\left(b_2-a_2\right)}\right)$$

$$c_4 = 0$$

Case 3: $0.5 > b_2$ and $a_2 > -0.5 > a_1$,

$$c_1 = (O-M)\left(\frac{2\sqrt{3}\left(4b_1^3-1-3b_1\right)}{24\left(b_1-a_1\right)}-\frac{2\sqrt{3}\left(4\left(b_2^3-a_2^3\right)-3\left(b_2-a_2\right)\right)}{24\left(b_2-a_2\right)}\right)$$

$$c_2 = (O-M)\left(\frac{2\sqrt{3}\left(4b_1^3-12a_1b_1^2-12a_1b_1-3a_1-3b_1-1\right)}{48\left(b_1-a_1\right)^2}-\frac{2\sqrt{3}\left(b_2-a_2\right)^2}{12\left(b_2-a_2\right)}\right)$$

$$c_3 = (O-M)\left(\frac{6\sqrt{5}\left(4{b_1}^2-1\right)^2}{192\left(b_1-a_1\right)}-\frac{6\sqrt{5}\left(2\left({b_2}^4-{a_2}^4\right)+{a_2}^2-{b_2}^2\right)}{24\left(b_2-a_2\right)}\right)$$

$$c_4 = (O-M)\left(\frac{\sqrt{5}\left(16{a_1}^2{b_1}^2+4{b_1}^2+16{a_1}^2b_1+1+4a_1+4b_1+4{a_1}^2+16a_1b_1+16a_1{b_1}^2\right)}{32\left(b_1-a_1\right)^3}\right)$$

Case 4: $b_1 > -0.5 > a_2$,

$$c_1 = (O-M)\left(\frac{2\sqrt{3}\left(4{b_1}^3-1-3b_1\right)}{24\left(b_1-a_1\right)}-\frac{2\sqrt{3}\left(4{b_2}^3-1-3b_2\right)}{24\left(b_2-a_2\right)}\right)$$

$$c_2 = (O-M)\left(\begin{array}{l}\dfrac{2\sqrt{3}\left(4{b_1}^3-12a_1{b_1}^2-12a_1b_1-3a_1-3b_1-1\right)}{48\left(b_1-a_1\right)^2}\\ -\dfrac{2\sqrt{3}\left(4{b_2}^3-12a_2{b_2}^2-12a_2b_2-3a_2-3b_2-1\right)}{48\left(b_2-a_2\right)^2}\end{array}\right)$$

$$c_3 = (O-M)\left(\frac{6\sqrt{5}\left(4{b_1}^2-1\right)^2}{192\left(b_1-a_1\right)}-\frac{6\sqrt{5}\left(4{b_2}^2-1\right)^2}{192\left(b_2-a_2\right)}\right)$$

$$c_4 = (O-M)\left(\begin{array}{l}\dfrac{\sqrt{5}\left(16{a_1}^2{b_1}^2+4{b_1}^2+16{a_1}^2b_1+1+4a_1+4b_1+4{a_1}^2+16a_1b_1+16a_1{b_1}^2\right)}{32\left(b_1-a_1\right)^3}\\ -\dfrac{\sqrt{5}\left(16{a_2}^2{b_2}^2+4{b_2}^2+16{a_2}^2b_2+1+4a_2+4b_2+4{a_2}^2+16a_2b_2+16a_2{b_2}^2\right)}{32\left(b_2-a_2\right)^3}\end{array}\right)$$

Case 5: $b_2 > 0.5$ and $a_2 > -0.5 > a_1$,

$$c_1 = (O-M)\left(\frac{2\sqrt{3}\left(4{b_1}^3-1-3b_1\right)}{24\left(b_1-a_1\right)}+\frac{2\sqrt{3}\left(4{b_2}^3-1-3b_2\right)}{24\left(b_2-a_2\right)}\right)$$

$$c_2 = (O-M)\left(\frac{2\sqrt{3}\left(4b_1^3 - 12a_1b_1^2 - 12a_1b_1 - 3a_1 - 3b_1 - 1\right)}{48(b_1 - a_1)^2} + \frac{2\sqrt{3}\left(4b_2^3 - 12a_2b_2^2 - 12a_2b_2 - 3a_2 - 3b_2 - 1\right)}{48(b_2 - a_2)^2}\right)$$

$$c_3 = (O-M)\left(\frac{6\sqrt{5}\left(4b_1^2 - 1\right)^2}{192(b_1 - a_1)} + \frac{6\sqrt{5}\left(4b_2^2 - 1\right)^2}{192(b_2 - a_2)}\right)$$

$$c_4 = (O-M)\left(\frac{\sqrt{5}\left(16a_1^2b_1^2 + 4b_1^2 + 16a_1^2b_1 + 1 + 4a_1 + 4b_1 + 4a_1^2 + 16a_1b_1 + 16a_1b_1^2\right)}{32(b_1 - a_1)^3} + \frac{\sqrt{5}\left(16a_2^2b_2^2 + 4b_2^2 + 16a_2^2b_2 + 1 + 4a_2 + 4b_2 + 4a_2^2 + 16a_2b_2 + 16a_2b_2^2\right)}{32(b_2 - a_2)^3}\right)$$

The ratios of the coefficients can be easily computed from these formulae.

References:

[1] Adelson, H., and Simoncelli, E., "Orthogonal pyramid transforms for image coding," *SPIE*, Vol. 845, 1987, pp. 50-58.

[2] Ahmed, N., Natarjan, T. and Rao, K. R., 'Discrete Cosine Transform', *IEE Trans Computers*, Vol23, 1974, pp90-93.

[3] Baharav, Z., Malah, D., and Karnin, E., "Hierarchical interpretation of fractal image coding and its applications to fast decoding," *Intl. Conf. on Digital Signal Processing*, Cyprus, July 1993.

[4] Barnsley, M. F. and Demko, S. G. in *Rational Approximation and Interpolation*, eds. Graves-Moris, P. R., Staff, E. B. and Varga, R. S., Springer, New York, pp. 73-88, 1984.

[5] Barnsley, M. F., Demko, S., Iterated function systems and the global construction of fractals, *Proc. R. Soc. Lond.* A 399, 243-275 (1985)

[6] Barnsley, M. F., Ervin, V., Hardin, D., and Lancaster, J., "Solution of an inverse problem for fractals and other sets," *Proc. Natl. Acad. Sci. USA*, Vol. 83, pp. 1975-1977, April 1986.

[7] Barthel, K. U., and Voyé, T., "Adaptive fractal image coding in the frequency domain," *Proceedings of International Workshop on Image Processing: Theory, Methodology, Systems and Applications*, Budapest, June 1994.

[8] Bethel, D., *Optimisation of Still Image Compression Techniques*, PhD Thesis, University of Bath, 1997.

[9] Bethel, D., and Monro, D. M., "Optimal parent pruning in fractal compression," *Proc. IEEE International Conference on Image Processing 97*, Santa Barbara, California, Oct. 1997.

[10] Boss, R. D., and Jacobs, E. W., "Archetype classification in an iterated transformation image compression algorithm," *Fractal Image Compression: Theory and Application*, Y. Fisher (ed.), Springer-Verlag, New York, 1994.

[11] Beaumont, J. M., "Image data compression using fractal techniques," British Telecom Technology Journal, Vol. 9, No. 4, pp. 93- 109, 1991.

[12] Cohen, A., Daubechies, I., and Feauveau, J. C., "Biorthogonal Bases of Compactly Supported Wavelets", *Commun. Pure Appl. Math*, Vol.45, 1992, pp485-560.

[13] Culik II, K., and Kari, J., "Inference algorithms for WFA and image compression," *Fractal Image Compression: Theory and Application*, Y. Fisher (ed.), SpringerVerlag, New York, 1994.

[14] Cvetkovic, Z. and Popovic, M. V., "New fast recursive algorithms for the computation of discrete cosine and sine transforms," *IEEE Transactions on Signal Processing*, Vol. 40, No. 8, August 1992.

[15] Daubechies, I., "Orthonormal bases of compactly supported wavelets", *Comm. Pure Appl. Math.*, Vol. 41, pp 909-966, 1988.

[16] Davis, G., "Adaptive self-quantization of wavelet subtrees: A wavelet based theory for fractal image compression," *SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing 95*, San Diego, July 1995.

[17] Dudbridge, F., *Image Approximation by Self Affine Fractals*, PhD thesis, University of London, 1992.

[18] Falconer, K. J., *Fractal Geometry: Mathematical Foundations and Applications*, John Wiley & Sons Ltd, England.

[19] Fisher, Y., *Fractal Image Compression: Theory and Application*, SpringerVerlag, New York, 1994.

[20] Fisher, Y., Jacobs, E. W., and Boss, R. D., "Fractal image compression using iterated transforms." In *Image and Text Compression*, chapter 2, pp. 35-61. J. A. Storer, editor, Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[21] Forte, B., and Vrscay, E. R., "Theory of generalized fractal transforms," in *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, Trondheim, July 1995, Y.Fisher (ed.), Springer-Verlag, Heidelberg, 1998.

[22] Frigaard, C., Gade, J., Hemmingsen, T., and Sand, T., "Image compression based on fractal theory," Manuscript, Institute for Electronic Systems, Aalborg University, 1994.

[23] Gharavi-Alkhansari, M., and Huang, T., "Fractal based techniques for a generalized image coding method," *Proc. IEEE International Conference on Image Processing 94*, Austin, Texas, Nov. 1994.

[24] Gharavi-Alkhansari, M., and Huang, T., "Fractal image coding using rate-distortion optimized matching pursuit," *Proc. SPIE VCIP 1996*, Vol. 2727, 1996.

[25] Gonzalez, R. C., and Woods, E. W., *Digital Image Processing*, Addison Wesley, 1992.

[26] Götting, D., Ibenthal, A., and Grigat, R. R., "Fractal image coding and magnification using invariant features," *NATO ASI Conf. Fractal Image Encoding and Analysis*, Trondheim, July 1995, appears in *Fractals*, Volume 5, Supplementary issue, April 1997.

[27] Hauske, G., "A self organizing map approach to image quality," BioSystems 40 (1997) 93-102.

[28] Huffman, D. A., "A method for construction of minimum redundancy codes," *Proc. IRE*, Vol. 40, 1962, pp1016-1021.

[29] Hürtgen, B., "Performance bounds for fractal coding", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 95*, Vol. 4, Detroit, 1995.

[30] Hürtgen, B., "Statistical evaluation of fractal coding schemes," *Proc. IEEE International Conference on Image Processing 95*, Washington, D.C., 1995.

[31] Hürtgen, B., Mols, P., and Simon, S. F., "Fractal transform coding of color images", *Proceedings from SPIE Visual Communications and Image Processing*, pp. 1683-1691, 1994.

[32] Hürtgen, B., and Stiller, C., "Fast hierarchical codebook search for fractal coding of still images," *EOS/SPIE Visual Communications and PACS for Medical Applications '93*, Berlin, 1993.

[33] Hutchinson, J. E., Fractals and self-similarity, *Indiana University Mathematics Journal*, 3(5):713-747, 1981

[34] Jain, A. K., "Image data compression: a review," *Proceedings of the IEEE*, Vol. 69, No. 3, March 1981.

[35] Jacquin, A. E., "A novel fractal block-coding technique for digital images", *Proc. ICASSP 90*, pp. 2225-2228, 1990.

[36] Jacquin, A. E., "Fractal coding: a review", *Proc. IEEE*, Vol. 81, No. 10, October 1993, pp. 1451-1465.

[37] Kominek, J., "Convergence of fractal encoded images," *Proceedings DCC'95 Data Compression Conference 95*, J. A. Storer and M. Cohn (eds.), IEEE Comp. Soc. Press, March 1995.

[38] Krupnik, H., Malah, D., and Karnin, E., "Fractal representation of images via the discrete wavelet transform," *IEEE 18th Conf. of Electrical Engineering*, Tel-Aviv, March 1995.

[39] Kreyszig, E., *Introductory Functional Analysis with Applications,* John Wiley & Sons, New York, NY, USA, 1978.

[40] Lin, H. and Venetsanopoulos, A. N., "Removing blocking effects of fractal compressed images using multichannel filtering," *Proceedings of 2nd International Workshop on Image and Signal Processing: Theory, Method, Systems and Applications,* Budapest, Nov. 1995.

[41] Lundheim, L. M., "A discrete framework for fractal signal modeling." In *Fractal Image Compression: Theory and Application,* Y. Fisher, editor, chapter 7, pp. 137-151. Springer-Verlag, New York, NY, USA, 1995.

[42] Mallat, S., "A theory for multiresolution signal decomposition: The wavelet representation", *IEEE Trans. Pattern Anal. Mach. Intel.*, Vol. 11, July 1989.

[43] Mandelbrot, B. B., *Les objets fractals: forme, hasard et dimension.* Paris: Flammarion.

[44] Mandelbrot, B. B., *The Fractal Geometry of Nature,* W. H. Freeman and Company, New York.

[45] Max, J., "Qunatizing for minimum Distortion," *IEEE Trans. Inform. Theory,* March 1960, pp7-12.

[46] Monro, D. M., "A hybrid fractal transform," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 93,* Vol. 5, pp. 169-172, 1993.

[47] Monro, D. M., "Class of fractal transforms", *Electronics Letters* 29,4 (1993) 362--363.

[48] Monro, D. M. and Dickson, G. J., "Zerotree coding of DCT coefficients," *Proc. ICIP-97,* Vol. 2, pp. 625-628.

[49] Monro, D. M., and Dudbridge, F., "Fractal approximation of image blocks," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 92*, Vol. 3, pp. 485-488, 1992.

[50] Monro, D. M., and Dudbridge, F., "Rendering algorithms for deterministic fractals," *IEEE Computer Graphics and Applications* 15,1 (1995) pp. 32-41.

[51] Monro, D. M., Dudbridge, F., and Wilson, A., "Deterministic rendering of self-affine fractals," IEE Colloquium on Fractal Techniques in Image Processing, IEE *Digest* No. 1990/171, 1990.

[52] Monro, D. M. and Wakefield, P. D., "Zooming with implicit fractals," *Proceedings of International Conference on Image Processing 1997*, Santa Barbara, California, Oct. 1997.

[53] Monro, D. M., and Woolley, S. J., Fractal image compression without searching, in: *Proceedings of ICASSP1994 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 5, Adelaide, 1994.

[54] Monro, D. M., and Woolley, S. J., "Rate/distortion in fractal compression: Order of transform and block symmetries," *Proceedings International Symposium on Speech, Image Processing and Neural Networks '94*, Vol. 1, pp 168-171, Hong Kong, April 1994.

[55] Nelson, M. and Gailly, J., *The Data Compression Book*, 2nd Edition, MRT Books, 1996, pp. 113-152

[56] Novak, M., "Attractor coding of images," *Proceedings of the International Picture Coding Symposium '93*, Lausanne, March 1993.

[57] Øien, G. E., Baharav, Z., Lepsøy, S., Malah, D., and Karnin, E., "A new improved collage theorem with applications to multiresolution fractal image

coding," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1994*, Adelaide, 1994.

[58] Øien, G. E., and Lepsøy, S., "A class of fractal image coders with fast decoder convergence", in: *Fractal Image Compression: Theory and Application*, Y. Fisher (ed.), Springer-Verlag, New York, 1994.

[59] Øien, G. E., Lepsøy, S., and Ramstad, T., "An inner product space approach to image coding by contractive transformations," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing '91*, pp. 2773--2776, 1991.

[60] Øien, G. E., Lepsøy, S., and Ramstad, T., "Reducing the complexity of a fractal-based image coder," *Proceedings of the 5th European Signal Processing Conference 1992*, pp. 1353-1356, 1992

[61] Peitgen, H.-O., Jürgen, H. and Saupe, D., *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, 1992

[62] Polidori, E., and Dugelay, J. L., "Zooming using IFS," *NATO ASI Conf. Fractal Image Encoding and Analysis*, Trondheim, July 1995, appears in *Fractals*, Volume 5, Supplementary issue, April 1997.

[63] Reusens, E., "Overlapped adaptive partitioning for image coding based on the theory of iterated function systems," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 94*, Adelaide, 1994.

[64] Reusens, E., "Partitioning complexity issue for iterated function systems based image coding," *Proceedings of the VIIth European Signal Processing Conference '94*, Edinburgh, Sept. 1994.

[65] Rinaldo, R., and Calvagno, G., "An image coding scheme using block prediction of the pyramid subband decomposition," *Proc. IEEE International Conference on Image Processing 94*, Austin, Texas, Nov. 1994.

[66] Rinaldo, R., and Calvagno, G., "Image coding by block prediction of multiresolution subimages," *IEEE Transactions on Image Processing.* 4,7 (1995) pp. 909-920.

[67] Said, A., and Pearlman, W.A., "A new Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology.*

[68] Saupe, D., "Fractal image compression via nearest neighbor search," *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, Trondheim, July 1995, Y. Fisher (ed.), Springer-Verlag, Heidelberg, 1998.

[69] Saupe, D., "The futility of square isometries in fractal image compression," *Proceedings IEEE International Conference on Image Processing '96*, Vol. I, pp 161-164, Lausanne, Switzerland, September 1996.

[70] Shapiro, J., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Trans. Signal Processing*, Vol. 41 No. 12, December 1993, pp3445-3462.

[71] Shoham, Y., and Gersho, A., "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE trans. Acoust. Speech Signal Processing*, Vol.36 No. 9, September 1988, pp1445-1453.

[72] Signes, J., "Geometrical interpretation of IFS based image coding", *NATO ASI Conf. Fractal Image Encoding and Analysis*, Trondheim, July 1995, appears in *Fractals*, Volume 5, Supplementary issue, April 1997.

[73] Vetterli, M. and Herley, C., "Wavelets and Filter Banks: Theory and Design", *IEEE Trans. Signal Processing*, Vol.40 No. 9, September 1992, pp2207-2232.

[74] Vines, G., "Orthogonal basis IFS", in: *Fractal Image Compression: Theory and Application*, Y. Fisher (ed.), Springer-Verlag, New York, 1994.

[75] Vrscay, E. R., "Mathematical theory of generalized fractal transforms and associated inverse problems," *Proc. ImageTech Conference on the Mathematics of Imaging and its Applications*, March 17-20, 1996, Atlanta, Georgia

[76] Wakefield, P. D., *Transfer Report on Fractal Image Compression*, University of Bath, 1997.

[77] Wakefield, P. D., Bethel, D. M., and Monro, D. M., "Hybrid image compression with implicit fractal terms," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1997*, Munich, 1997.

[78] Wallace, G. K., 'Overview of the JPEG (ISO/CCITT) Still Image Compression Standard', Image Processing Algorithms and Techniques, *Proc. SPIE*, Vol 1244, February 1990, pp 220-233.

[79] van der Walle, A., "Merging fractal image compression and wavelet transform methods," *NATO ASI Conf. Fractal Image Encoding and Analysis*, Trondheim, July 1995, appears in *Fractals*, Volume 5, Supplementary issue, April 1997.

[80] Wohlberg, B. E., *Fractal Image Compression and the Self-Affinity Assumption: A Stochastic Modelling Perspective*, PhD Thesis, University of Cape Town, 1996.

[81] Woolley, S. J., *Optimising Polynomial Fractal Transforms for Image Compression*, PhD Thesis, University of Bath, 1996.

[82] Woolley, S. J., and Monro, D. M., "Rate/distortion performance of fractal transforms for image compression," *Fractals*, 2(3):395-398, 1994.

[83] Woolley, S. J., and Monro, D. M., Optimum parameters for hybrid fractal image coding, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 95*, Detroit, 1995.

[84] Xu, W., and Hauske, G., "Comparison and optimization of adaptive and nonadaptive subband image coding techniques," *Proc. PCS '93*, Lausanne, Switzerland, 7.1, 1993

[85] Xu, W., and Hauske, G., "Picture quality evaluation based on error segmentation," *Proc. SPIE*, vol. 208, *Visual Communications and Image Processing '94*, Chicago, 1994.